(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2012 年 10 月 4 日(04.10.2012)

WIPO | PCT

(10) 国際公開番号
WO 2012/132623 A1

(51) 国際特許分類:
*H04L 9/06* (2006.01) *G09C 1/00* (2006.01)

(21) 国際出願番号: PCT/JP2012/053933

(22) 国際出願日: 2012 年 2 月 20 日(20.02.2012)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願 2011-069185 2011 年 3 月 28 日(28.03.2011) JP
特願 2011-207705 2011 年 9 月 22 日(22.09.2011) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社(SONY CORPORATION) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 渋谷 香士(SHIBUTANI, Kyoji) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 ソニー株式会社内 Tokyo (JP). 三津田 敦司(MITSUDA, Atsushi) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 ソニー株式会社内 Tokyo (JP). 秋下 徹(AKISHITA, Toru) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 ソニー株式会社内 Tokyo (JP). 五十部 孝典(ISOBE, Takanori) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 ソニー株式会社内 Tokyo (JP). 白井 太三(SHIRAI, Taizo) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 ソニー株式会社内 Tokyo (JP). 樋渡 玄良(HIWATARI, Harunaga) [JP/JP]; 〒1080075 東京都港区港南１丁目７番１号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 宮田 正昭, 外(MIYATA, Masaaki et al.); 〒1040032 東京都中央区八丁堀三丁目２５番９号 ＫＳＫビル西館８階 特許業務法人 大同特許事務所 Tokyo (JP).

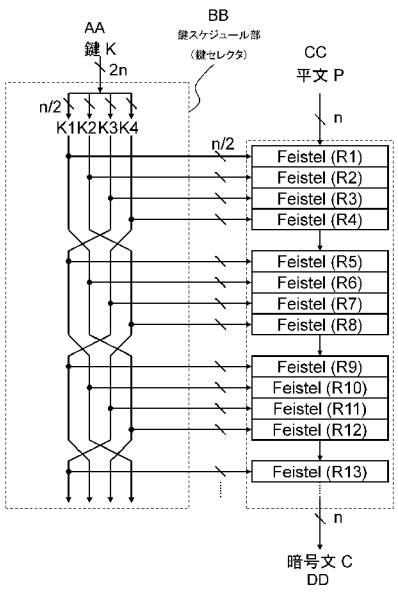
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA,

[続葉有]

(54) **Title**: ENCRYPTION PROCESSING DEVICE, ENCRYPTION PROCESSING METHOD, AND PROGRAMME

(54) 発明の名称 : 暗号処理装置、および暗号処理方法、並びにプログラム

[図21]




AA Key (K)
BB Key schedule unit (key selector)
CC Plain text (P)
DD Encrypted text (C)

(57) **Abstract**: In order to make an encryption processing device which is very safe due to the control of the supply of a round key, one aspect of this invention comprises: an encryption processing unit which divides configuration bits of data which is to undergo data processing into a plurality of lines and then inputs the same, and then repeatedly executes, as a round operation, a data conversion process, to which a round function is applied, on the data of each line; and a key schedule unit which outputs a round key to a round operation execution unit in the encryption processing unit. The key schedule unit is a permutation-type key schedule unit which divides previously held secret keys into a plurality of units and generates a plurality of round keys, and the key schedule unit outputs the plurality of generated round keys in a setting in which there is no repetition of a fixed sequence, to the round operation execution units sequentially executed in the encryption processing unit. Due to said structure, an encryption processing configuration which is very safe and is strongly resistant to related-key attack or similar is achieved.

(57) 要約: ラウンド鍵の供給制御による安全性の高い暗号処理装置を実現する。データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して複数のラウンド鍵を生成する置換型鍵スケジュール部であり、暗号処理部において順次実行するラウンド演算実行部に対して、生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する。本構成により、例えば関連鍵攻撃等に対する耐性の高い安全性の高い暗号処理構成が実現される。

WO 2012/132623 A1

RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告（条約第 21 条(3)）

— 補正された請求の範囲（条約第 19 条(1)）

# 明　細　書

## 発明の名称 ：

## 暗号処理装置、および暗号処理方法、並びにプログラム

## 技術分野

[0001] 本開示は、暗号処理装置、および暗号処理方法、並びにプログラムに関する。さらに詳細には、共通鍵系暗号を実行する暗号処理装置、および暗号処理方法、並びにプログラムに関する。

## 背景技術

[0002] 情報化社会が発展すると共に、扱う情報を安全に守るための情報セキュリティ技術の重要性が増してきている。情報セキュリティ技術の構成要素の一つとして暗号技術があり、現在では様々な製品やシステムで暗号技術が利用されている。

[0003] 暗号処理アルゴリズムには様々なものがあるが、基本的な技術の一つとして、共通鍵ブロック暗号と呼ばれるものがある。共通鍵ブロック暗号では、暗号化用の鍵と復号用の鍵が共通のものとなっている。暗号化処理、復号処理共に、その共通鍵から複数の鍵を生成し、あるブロック単位、例えば６４ビット、１２８ビット、２５６ビット等のブロックデータ単位でデータ変換処理を繰り返し実行する。

[0004] 代表的な共通鍵ブロック暗号のアルゴリズムとしては、過去の米国標準であるＤＥＳ（Ｄａｔａ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）や現在の米国標準であるＡＥＳ（Ａｄｖａｎｃｅｄ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）が知られている。他にも様々な共通鍵ブロック暗号が現在も提案され続けており、２００７年にソニー株式会社が提案したＣＬＥＦＩＡも共通鍵ブロック暗号の一つである。

[0005] このような、共通鍵ブロック暗号のアルゴリズムは、主として、入力データの変換を繰り返し実行するラウンド関数実行部を有する暗号処理部と、ラウンド関数部の各ラウンドで適用するラウンド鍵を生成する鍵スケジュール

部とによって構成される。鍵スケジュール部は、秘密鍵であるマスター鍵（主鍵）に基づいて、まずビット数を増加させた拡大鍵を生成し、生成した拡大鍵に基づいて、暗号処理部の各ラウンド関数部で適用するラウンド鍵（副鍵）を生成する。

[0006] このようなアルゴリズムを実行する具体的な構造として、線形変換部および非線形変換部を有するラウンド関数を繰り返し実行する構造が知られている。例えば代表的な構造にＦｅｉｓｔｅｌ構造や一般化Ｆｅｉｓｔｅｌ構造がある。Ｆｅｉｓｔｅｌ構造や一般化Ｆｅｉｓｔｅｌ構造は、データ変換関数としてのＦ関数を含むラウンド関数の単純な繰り返しにより、平文を暗号文に変換する構造を持つ。Ｆ関数においては、線形変換処理および非線形変換処理が実行される。なお、Ｆｅｉｓｔｅｌ構造を適用した暗号処理について記載した文献としては、例えば非特許文献１、非特許文献２がある。

[0007] 一方、暗号アルゴリズムの解析や、鍵の解析などを不正に実行して暗号解読を行う手法も様々な新しい手法が出現している。その１つとして、例えば、関連鍵攻撃がある。このような攻撃に対応するために様々な対策も考えられているものの、十分なものとは言えないのが現状である。

## 先行技術文献

### 非特許文献

[0008] 非特許文献1：Ｋ．　Ｎｙｂｅｒｇ，　”Ｇｅｎｅｒａｌｉｚｅｄ　Ｆｅｉｓｔｅｌ　ｎｅｔｗｏｒｋｓ”，　ＡＳＩＡＣＲＹＰＴ　９６，　ＳｐｒｉｎｇｅｒＶｅｒｌａｇ，　１９９６，　ｐｐ．９１－－１０４．

非特許文献2：Ｙｕｌｉａｎｇ　Ｚｈｅｎｇ，　Ｔｓｕｔｏｍｕ　Ｍａｔｓｕｍｏｔｏ，　Ｈｉｄｅｋｉ　Ｉｍａｉ：　Ｏｎ　ｔｈｅ　Ｃｏｎｓｔｒｕｃｔｉｏｎ　ｏｆ　Ｂｌｏｃｋ　Ｃｉｐｈｅｒｓ　Ｐｒｏｖａｂｌｙ　Ｓｅｃｕｒｅ　ａｎｄ　Ｎｏｔ　Ｒｅｌｙｉｎｇ　ｏｎ　Ａｎｙ　Ｕｎｐｒｏｖｅｄ　Ｈｙｐｏｔｈｅｓｅｓ．　ＣＲＹＰＴＯ　１９８９：　４６１－４８０

非特許文献3：Ｓｏｎｙ　Ｃｏｒｐｏｒａｔｉｏｎ，　”Ｔｈｅ　１２８－ｂｉｔ　Ｂｌｏｃｋｃｉｐｈｅｒ　ＣＬＥＦＩＡ　Ａｌｇｏｒｉｔｈｍ　Ｓｐ

ｅｃｉｆｉｃａｔｉｏｎ”，　Ｒｅｖｉｓｉｏｎ　１．０，　２００７．

非特許文献4：青木，市川，神田，松井，盛合，中嶋，時田，”１２８ビットブロック暗号　Ｃａｍｅｌｌｉａ　アルゴリズム仕様書”，第　２．０版，２００１

非特許文献5：ＧＯＳＴ　２８１４７－８９：　Ｅｎｃｒｙｐｔｉｏｎ，　Ｄｅｃｒｙｐｔｉｏｎ，　ａｎｄ　Ｍｅｓｓａｇｅ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｃｏｄｅ　（ＭＡＣ）　Ａｌｇｏｒｉｔｈｍｓ，　ＲＦＣ　５８３０．

非特許文献6：３ｒｄ　Ｇｅｎｅｒａｔｉｏｎ　Ｐａｒｔｎｅｒｓｈｉｐ　Ｐｒｏｊｅｃｔ，　Ｔｅｃｈｎｉｃａｌ　Ｓｐｅｃｉｆｉｃａｔｉｏｎ　Ｇｒｏｕｐ　Ｓｅｒｖｉｃｅｓ　ａｎｄ　Ｓｙｓｔｅｍ　Ａｓｐｅｃｔｓ，　３Ｇ　Ｓｅｃｕｒｉｔｙ，　Ｓｐｅｃｉｆｉｃａｔｉｏｎ　ｏｆ　ｔｈｅ　３ＧＰＰ　Ｃｏｎｆｉｄｅｎｔｉａｌｉｔｙ　ａｎｄ　Ｉｎｔｅｇｒｉｔｙ　Ａｌｇｏｒｉｔｈｍｓ；　Ｄｏｃｕｍｅｎｔ　２：　ＫＡＳＵＭＩ　Ｓｐｅｃｉｆｉｃａｔｉｏｎ，　Ｖ９．０．０，　２００９

## 発明の概要

## 発明が解決しようとする課題

[0009] 本開示は、例えば上述の状況に鑑みてなされたものであり、関連鍵攻撃等の不正な暗号解読を困難化し、安全性の高い暗号処理装置、および暗号処理方法、並びに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

[0010] 本開示の第１の側面は、

データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、

前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する暗号処理装置にある。

[0011] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、前記ラウンド演算実行部に対する前記複数のラウンド鍵の入力シーケンスを、前記ラウンド演算実行部における複数ラウンド単位で変更する。

[0012] さらに、本開示の暗号処理装置の一実施態様において、前記暗号処理部は、前記複数のラインに分割されたデータを入力して非線形変換処理と、線形変換処理を含むＦ関数実行部と、前記Ｆ関数実行部の出力に対して、前記ラウンド鍵を適用した演算を実行する演算部を有する。

[0013] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵と同一のビット数を持つ複数のラウンド鍵を生成する。

[0014] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵より少ないビット数を持つ複数のラウンド鍵構成データを生成し、前記複数のラウンド鍵構成データを複数連結して前記ラウンド演算実行部に入力するラウンド鍵と同一ビット数のラウンド鍵を生成する。

[0015] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、前記暗号処理部において順次実行するラウンド演算実行部に対して、ラウンド演算実行部において並列適用する複数のラウンド鍵を並列出力する。

[0016] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、前記ラウンド演算実行部に対する鍵の選択供給処理を行う１以上のセレクタを有する。

[0017] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、設定したグループ単位で、前記ラウンド演算実行部に対する鍵供給シーケンスの制御処理を行う。

[0018] さらに、本開示の暗号処理装置の一実施態様において、前記鍵スケジュール部は、前記グループ単位のセレクタを有する。

[0019] さらに、本開示の暗号処理装置の一実施態様において、前記暗号処理部は、入力データとしての平文を暗号文に変換する暗号化処理、または、入力データとしての暗号文を平文に変換する復号処理を実行する。

[0020] さらに、本開示の第２の側面は、

暗号処理装置において実行する暗号処理方法であり、

暗号処理部が、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理ステップと、

鍵スケジュール部が、前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジューリングステップを実行し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する暗号処理方法にある。

[0021] さらに、本開示の第３の側面は、

暗号処理装置において暗号処理を実行させるプログラムであり、

暗号処理部に、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行させる暗号処理ステップと

、

鍵スケジュール部に、前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジューリングステップを実行させ、

前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記鍵スケジュール部に、前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力させるプログラムにある

[0022] なお、本開示のプログラムは、例えば、様々なプログラム・コードを実行可能な情報処理装置やコンピュータ・システムに対して例えば記憶媒体によって提供されるプログラムである。このようなプログラムを情報処理装置やコンピュータ・システム上のプログラム実行部で実行することでプログラムに応じた処理が実現される。

[0023] 本開示のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

## 発明の効果

[0024] 本開示の一実施例によれば、ラウンド鍵の供給制御により安全性の高い暗号処理装置が実現される。

具体的には、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して複数のラウンド鍵を生成する置換型鍵スケジュール部であり、暗号処理部におい

て順次実行するラウンド演算実行部に対して、生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する。本構成により、例えば関連鍵攻撃等に対する耐性の高い安全性の高い暗号処理構成が実現される。

## 図面の簡単な説明

[0025] [図1]ｋビットの鍵長に対応したｎビット共通鍵ブロック暗号アルゴリズムを説明する図である。

[図2]図１に示すｋビットの鍵長に対応したｎビット共通鍵ブロック暗号アルゴリズムに対応する復号アルゴリズムを説明する図である。

[図3]鍵スケジュール部とデータ暗号化部の関係について説明する図である。

[図4]データ暗号化部の構成例について説明する図である。

[図5]ＳＰＮ構造のラウンド関数の例について説明する図である。

[図6]Ｆｅｉｓｔｅｌ構造のラウンド関数の一例について説明する図である。

[図7]拡張Ｆｅｉｓｔｅｌ構造の一例について説明する図である。

[図8]拡張Ｆｅｉｓｔｅｌ構造の一例について説明する図である。

[図9]非線形変換部の構成例について説明する図である。

[図10]線形変換処理部の構成例について説明する図である。

[図11]鍵スケジュール部（拡大鍵生成部）について説明する図である。

[図12]鍵スケジュール部（拡大鍵生成部）について説明する図である。

[図13]ブロック暗号ＧＯＳＴの構成例について説明する図である。

[図14]ＧＯＳＴのデータ構造であるＦｅｉｓｔｅｌ構造について説明する図である。

[図15]任意の秘密鍵差分Δを与えることのできる攻撃について説明する図である。

[図16]置換型鍵スケジュール部を持つ場合に、各ラウンド鍵差分$\Delta_i$は秘密鍵差分Δをｍ－ｂｉｔ毎に分割したものとなることを説明する図である。

[図17]平文差分として（ｄ，ｄ）を与えることでＦ関数への入力差分が０となることについて説明する図である。

[図18]ｔｙｐｅ－２一般化Ｆｅｉｓｔｅｌ構造の例について説明する図である。

[図19]Ｆ関数の後で排他的論理和を取る構成例について説明する図である。

[図20]Ｆ関数の後で排他的論理和を取る構成例について説明する図である。

[図21]秘密鍵Ｋが２ｎ－ｂｉｔの場合（ｎ／２）－ｂｉｔ毎に４等分し、４ラウンド毎に入れ替えながら供給する構成例について説明する図である。

[図22]秘密鍵Ｋを（ｎ／４）－ｂｉｔ毎に８等分し、それらを４ラウンド毎に入れ替えながら、（ｎ／４）－ｂｉｔのものを２つずつ供給していく構成例について説明する図である。

[図23]秘密鍵Ｋの長さが４ｎ－ｂｉｔの場合、（ｎ／２）－ｂｉｔ毎に８等分し、８ラウンド毎に入れ替えながら供給していく構成例について説明する図である。

[図24]秘密鍵Ｋが２ｎ－ｂｉｔの場合、（ｎ／４）－ｂｉｔ毎に８等分し、４ラウンド毎に入れ替えながら（ｎ／４）－ｂｉｔのものを２つずつ供給していく構成例について説明する図である。

[図25]秘密鍵を（ｎ／４）－ｂｉｔ毎８等分した中で、４つを左側のＦ関数へのラウンド鍵$ＲＫ_{r,0}$に用いるものとし、残りの４つを右側のＦ関数へのラウンド鍵$ＲＫ_{r,1}$に用いる構成例を示す図である。

[図26]４系列一般化Ｆｅｉｓｔｅｌ構造におけるｃｙｃｌｉｃ　ｓｈｉｆｔをｒｏｕｎｄ　ｐｅｒｍｕｔａｔｉｏｎに変更したモデル（４系列一般化Ｆｅｉｓｔｅｌ構造＋）について説明する図である。

[図27]４系列一般化Ｆｅｉｓｔｅｌ構造に対して、図２５と同様の入れ替え手法を適用した例を説明する図である。

[図28]秘密鍵が（５ｎ／４）－ｂｉｔであり、１ラウンドに必要なラウンド鍵が（ｎ／２）－ｂｉｔのときの鍵スケジュール部の構成法を示す図である。

[図29]ラウンド鍵を単純に順番に入れる構成例を説明する図である。

[図30]ラウンド鍵を単純に順番に入れる構成例を説明する図である。

[図31]分割数ｄとしたｄ系列一般化Ｆｅｉｓｔｅｌ構造をもつｎ－ｂｉｔブロック暗号における１ラウンド辺りのラウンド鍵の利用構成について説明する図である。

[図32]秘密鍵がｎ－ｂｉｔの場合に、ｎ－ｂｉｔ秘密鍵を４等分したｎ／４－ｂｉｔのラウンド鍵を生成して各ラウンドに２つずつ入力する構成例を説明する図である。

[図33]秘密鍵が（５／４）ｎ－ｂｉｔの場合において、ｎ－ｂｉｔ秘密鍵を５等分したｎ／４－ｂｉｔのラウンド鍵を生成して各ラウンドに２つずつ入力する構成例を説明する図である。

[図34]秘密鍵が（５／４）ｎ－ｂｉｔの場合において、ｎ－ｂｉｔ秘密鍵を５等分したｎ／４－ｂｉｔのラウンド鍵を生成して各ラウンドに２つずつ入力する構成例を説明する図である。

[図35]複数ラウンド単位で鍵の選択順序を変更する構成例を説明する図である。

[図36]４ラウンド単位で実行する置換処理を毎回異なる態様に設定したラウンド鍵供給構成例を説明する図である。

[図37]複数ラウンド単位で鍵の選択順序を変更する構成例を説明する図である。

[図38]３ラウンド毎に置換を行い３ラウンドに６個の鍵を利用する構成例を説明する図である。

[図39]複数ラウンド単位で鍵の選択順序を変更する構成例を説明する図である。

[図40]ラウンド鍵供給構成としてのセレクタについて説明する図である。

[図41]ラウンド鍵供給構成としてのセレクタについて説明する図である。

[図42]ラウンド鍵供給構成としてのセレクタについて説明する図である。

[図43]暗号処理装置としてのＩＣモジュール７００の構成例を示す図である。

## 発明を実施するための形態

[0026] 以下、図面を参照しながら本開示に係る暗号処理装置、および暗号処理方法、並びにプログラムの詳細について説明する。説明は、以下の項目に従って行う。

１．共通鍵ブロック暗号の概要

２．鍵スケジュール部の構成と処理の概要について

３．鍵スケジュール部に対する攻撃とこれまでの対策例について

４．関連鍵攻撃に対し安全性を得られる置換型鍵スケジュール部について

５．鍵置換型鍵スケジュール部の様々な構成例（バリエーション）について

６．暗号処理装置の構成例について

７．本開示の構成のまとめ

[0027] ［１．共通鍵ブロック暗号の概要］

まず、共通鍵ブロック暗号の概要について説明する。

（１－１．共通鍵ブロック暗号）

ここでは共通鍵ブロック暗号（以下ではブロック暗号）としては以下に定義するものを指すものとする。

ブロック暗号は入力として平文Ｐと鍵Ｋを取り、暗号文Ｃを出力する。平文と暗号文のビット長をブロックサイズと呼びここではｎで著す。ｎは任意の整数値を取りうるが、通常、ブロック暗号アルゴリズムごとに、あらかじめひとつに決められている値である。ブロック長がｎのブロック暗号のことをｎビットブロック暗号と呼ぶこともある。

[0028] 鍵のビット長をｋで表す。鍵は任意の整数値を取りうる。共通鍵ブロック暗号アルゴリズムは１つまたは複数の鍵サイズに対応することになる。例えば、あるブロック暗号アルゴリズムＡはブロックサイズｎ＝１２８であり、ｋ＝１２８またはｋ＝１９２またはｋ＝２５６の鍵サイズに対応するという構成もありうるものとする。

平文Ｐ：ｎビット

暗号文Ｃ：ｎビット

鍵Ｋ：ｋビット

[0029] 図１にｋビットの鍵長に対応したｎビット共通鍵ブロック暗号アルゴリズムＥの図を示す。

暗号化アルゴリズムＥに対応する復号アルゴリズムＤは暗号化アルゴリズムＥの逆関数$E^{-1}$と定義でき、入力として暗号文Ｃと鍵Ｋを受け取り，平文Ｐを出力する。図２に図１に示した暗号アルゴリズムＥに対応する復号アルゴリズムＤの図を示す。

[0030] （１－２．内部構成）

ブロック暗号は２つの部分に分けて考えることができる。ひとつは鍵Ｋを入力とし、

ある定められたステップによりビット長を拡大してできた拡大鍵Ｋ’（ビット長ｋ’）を出力する「鍵スケジュール部」と、もうひとつは平文Ｐと鍵スケジュール部から拡大された鍵Ｋ’を受け取ってデータの変換を行い暗号文Ｃを出力する「データ暗号化部」である。

２つの部分の関係は図３に示される。

[0031] （１－３．データ暗号化部）

以下の実施例において用いるデータ暗号化部はラウンド関数という処理単位に分割できるものとする。ラウンド関数は入力としての２つのデータを受け取り、内部で処理を施したのち、１つのデータを出力する。入力データの一方は暗号化途中のｎビットデータであり、あるラウンドにおけるラウンド関数の出力が次のラウンドの入力として供給される構成となる。もう１つの入力データは鍵スケジュールから出力された拡大鍵の一部のデータが用いられ、この鍵データのことをラウンド鍵と呼ぶものとする。またラウンド関数の総数は総ラウンド数と呼ばれ、暗号アルゴリズムごとにあらかじめ定められている値である。ここでは総ラウンド数をＲで表す。

[0032] データ暗号化部の入力側から見て１ラウンド目の入力データを$X_1$とし、ｉ番目のラウンド関数に入力されるデータを$X_i$、ラウンド鍵を$RK_i$とすると、データ暗号化部全体は図４のように示される。

[0033] （１－４．ラウンド関数）

ブロック暗号アルゴリズムによってラウンド関数はさまざまな形態をとりうる。ラウンド関数はその暗号アルゴリズムが採用する構造（ｓｔｒｕｃｔｕｒｅ）によって分類できる。代表的な構造としてここではＳＰＮ構造、Ｆｅｉｓｔｅｌ構造、拡張Ｆｅｉｓｔｅｌ構造を例示する。

[0034] （ア）ＳＰＮ構造ラウンド関数

ｎビットの入力データすべてに対して、ラウンド鍵との排他的論理和演算、非線形変換、線形変換処理などが適用される構成。各演算の順番は特に決まっていない。図５にＳＰＮ構造のラウンド関数の例を示す。

[0035] （イ）Ｆｅｉｓｔｅｌ構造

ｎビットの入力データはｎ／２ビットの２つのデータに分割される。うち片方のデータとラウンド鍵を入力として持つ関数（Ｆ関数）が適用され、出力がもう片方のデータに排他的論理和される。そののちデータの左右を入れ替えたものを出力データとする。Ｆ関数の内部構成にもさまざまなタイプのものがあるが、基本的にはＳＰＮ構造同様にラウンド鍵データとの排他的論理和演算、非線形演算、線形変換の組み合わせで実現される。図６にＦｅｉｓｔｅｌ構造のラウンド関数の一例を示す。

[0036] （ウ）拡張Ｆｅｉｓｔｅｌ構造

拡張Ｆｅｉｓｔｅｌ構造はＦｅｉｓｔｅｌ構造ではデータ分割数が２であったものを，３以上に分割する形に拡張したものである。分割数をｄとすると、ｄによってさまざまな拡張Ｆｅｉｓｔｅｌ構造を定義することができる。Ｆ関数の入出力のサイズが相対的に小さくなるため、小型実装に向いているとされる。図７にｄ＝４でかつ、ひとつのラウンド内に２つのＦ関数が並列に適用される場合の拡張Ｆｅｉｓｔｅｌ構造の一例を示す。また，図８にｄ＝８でかつ，ひとつのラウンド内に１つのＦ関数が適用される場合の拡張Ｆｅｉｓｔｅｌ構造の一例を示す。

[0037] （エ）ｄ系列一般化Ｆｅｉｓｔｅｌ構造

分割数ｄが偶数である拡張Ｆｅｉｓｔｅｌ構造において、ひとつのラウン

ド内にｄ／２個のＦ関数が並列に適用される。

また、ラウンド間の置換としてｃｙｃｌｉｃ　ｓｈｉｆｔを用いる。

なお、図７、１８、２０は４系列一般化Ｆｅｉｓｔｅｌ構造を示す。

[0038] （１－５．非線形変換処理部）

非線形変換処理部は、入力されるデータのサイズが大きくなると実装上のコストが高くなる傾向がある。それを回避するために対象データを複数の単位に分割し、それぞれに対して非線形変換を施す構成がとられることが多い。例えば入力サイズをｍｓビットとして、それらをｓビットずつのｍ個のデータに分割して、それぞれに対してｓビット入出力を持つ非線形変換を行う構成である。それらのｓビット単位の非線形変換をＳ－ｂｏｘと呼ぶものとする。図９に例を示す。

[0039] （１－６．線形変換処理部）

線形変換処理部はその性質上、行列として定義することが可能である。行列の要素はＧＦ（$2^8$）の体の要素やＧＦ（２）の要素など、一般的にはさまざまな表現ができる。図１０にｍｓビット入出力をもち、ＧＦ（$2^s$）の上で定義されるｍ×ｍの行列により定義される線形変換処理部の例を示す。

[0040] ［２．鍵スケジュール部の構成と処理の概要について］

本開示の暗号処理の説明の前に、前提となる構成である鍵スケジュール部の構成と処理の概要について説明する。

[0041] 鍵スケジュール部（拡大鍵生成部）は、図１１に示すように、ｋ－ｂｉｔの秘密鍵Ｋを入力として、ある定められた変換によりｋ’－ｂｉｔの拡大鍵（ラウンド鍵）Ｋ’を生成する関数である。

[0042] 一般的にはｋ＜ｋ’であり、データ暗号化部がラウンド関数と呼ばれる非線形演算の繰り返し処理を行う場合、各ラウンド関数に供給するラウンド鍵をｍ－ｂｉｔ、ラウンド関数の繰り返し回数をＲとしたとき、ｋ’＝ｍ×Ｒとなる。図１２に示すような設定である。

[0043] 例えば、通鍵ブロック暗号ＡＥＳにおいては、

ｋ＝１２８の場合、

ｍ＝１２８，Ｒ＝１１であるため、

ｋ’＝１４０８である。

[0044] ｋ＝１９２の場合、

ｍ＝１２８，Ｒ＝１３であるため、

ｋ’＝１６６４である。

[0045] 鍵スケジュール部には安全性上以下のような特性（特性１～３）が求められる。

（特性１）等価鍵が存在しないこと

なお、秘密鍵Ｋ０を入力したときの拡大鍵Ｋ０’と，秘密鍵Ｋ１（≠Ｋ０）を入力したときの拡大鍵Ｋ１’に対し、Ｋ０’＝Ｋ１’が成り立つとき、Ｋ０とＫ１を等価鍵と呼ぶ。

[0046] （特性２）関連鍵攻撃（ｒｅｌａｔｅｄ　ｋｅｙ　ａｔｔａｃｋ）に対して十分な耐性を持つこと

一般的な攻撃は、ある固定された秘密鍵上での平文や暗号文間のデータの偏り（差分，線形性など）を利用したものであるのに対し、関連鍵攻撃では複数の秘密鍵上での平文や暗号文間データの偏りを利用するものである。

[0047] 例えば、２つの秘密鍵の値は分からないが、秘密鍵の差分については既知であるという仮定のもとで行う攻撃である。

また、秘密鍵の差分を攻撃者が自由に選べるという強力な仮定を置く攻撃も存在する。この仮定の下でも安全性を達成していることが望ましい。

[0048] 例えば、通常の差分攻撃（関連鍵を考慮しない差分攻撃）の場合、攻撃者は未知の秘密鍵Ｋによって暗号化された複数の平文Ｐ，暗号文Ｃ（＝Ｅ（Ｋ，Ｐ））の組のみを利用して、その未知の秘密鍵Ｋを導出する。

[0049] しかしながら、関連鍵差分攻撃の場合、攻撃者は未知の秘密鍵Ｋによって暗号化された複数の平文Ｐ，暗号文Ｃ（＝Ｅ（Ｋ，Ｐ））の組に加えて、その未知の秘密鍵Ｋに対し、攻撃者が指定した任意の秘密鍵差分ΔＫを加えたＫ（＋）ΔＫによって暗号化された複数の平文Ｐ’，暗号文Ｃ’（＝Ｅ（Ｋ（＋）ΔＫ，Ｐ’））の組も用いて未知の秘密鍵Ｋを導出する。

但し、（＋）は排他的論理和演算子を表す。

よって、関連鍵差分攻撃は攻撃者が利用できる情報がより増えているため通常の差分攻撃よりさらに強力な攻撃となる。

[0050] （特性３）スライド攻撃（ｓｌｉｄｅ　ａｔｔａｃｋ）に対し十分な耐性を持つこと

例えば、秘密鍵Ｋ０を入力したときのラウンド鍵を$RK_1$，$RK_2$，…，$RK_R$とし、秘密鍵Ｋ１を入力したときのラウンド鍵が$RK_2$，$RK_3$，…，$RK_{R+1}$のように、スライドした値を取るとき、安全性が損なわれる可能性がある。

[0051] さらに鍵スケジュール部には実装上以下のような特性（特性４～９）も要求される。

（特性４）実装が容易であること

（特性５）秘密鍵から拡大鍵を生成するセットアップ時間が短いこと

（特性６）ｏｎ－ｔｈｅ－ｆｌｙで拡大鍵が生成できること

（特性７）暗号化鍵スケジュール部，復号鍵スケジュール部ができる限り共有可能であること

（特性８）データ暗号化部，データ復号部とできる限り共有可能であること

（特性９）秘密鍵長の変更に容易に対応できること

[0052] 鍵スケジュール部は安全性、および実装の観点からこれらの特性をバランスよく満たすことが望まれる。

[0053] ［３．鍵スケジュール部に対する攻撃とこれまでの対策例について］

次に、鍵スケジュール部に対する攻撃とこれまでの対策例について説明する。

鍵スケジュール部は各種の共通鍵ブロック暗号毎に、様々な設計がなされている。

鍵スケジュール部の安全性、実装性能はデータ暗号化部と深い関連があるが、一般的にデータ暗号化部と同様に安全性と実装性能にはトレードオフが

あると考えられている。

[0054] 例えば、鍵スケジュール部に複雑な非線形関数を導入している暗号としてＣＬＥＦＩＡ（非特許文献３：Ｓｏｎｙ Ｃｏｒｐｏｒａｔｉｏｎ, ”Ｔｈｅ １２８－ｂｉｔ Ｂｌｏｃｋｃｉｐｈｅｒ ＣＬＥＦＩＡ Ａｌｇｏｒｉｔｈｍ Ｓｐｅｃｉｆｉｃａｔｉｏｎ”, Ｒｅｖｉｓｉｏｎ １．０, ２００７．）、 Ｃａｍｅｌｌｉａ（非特許文献４：青木，市川，神田，松井，盛合，中嶋，時田，”１２８ビットブロック暗号 Ｃａｍｅｌｌｉａ アルゴリズム仕様書”，第 ２．０版， ２００１）などが挙げられる。これらは関連鍵攻撃など鍵スケジュール部に起因する攻撃法に対して高い安全性を持つが、実装コストが比較的高く、特にハードウェア実装の際に回路規模が増大してしまうという問題がある。また、実装性能を高めるために比較的シンプルな非線形関数を導入した鍵スケジュール部を持つ暗号としてＡＥＳが挙げられるが、ＡＥＳは関連鍵攻撃に対して脆弱なことが知られている（１９２／２５６－ｂｉｔ鍵の場合）。

[0055] さらに実装性能を高める手法として非線形関数を用いずに線形関数のみで構成する手法が挙げられる。その中でも秘密鍵データを複数個に分割し、それらを置換するだけの鍵スケジュール部は、特にハードウェア実装の際に回路がほとんど不要であり、高い実装性能を持つとされる。このような鍵スケジュール部を置換型鍵スケジュール部と呼ぶ。

[0056] 置換型鍵スケジュール部は高い実装性能を持つものの、安全性上の問題を抱える方式が多い。

例えば、図１３に示すように、ブロック暗号ＧＯＳＴ（非特許文献５：ＧＯＳＴ ２８１４７－８９： Ｅｎｃｒｙｐｔｉｏｎ, Ｄｅｃｒｙｐｔｉｏｎ, ａｎｄ Ｍｅｓｓａｇｅ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｃｏｄｅ （ＭＡＣ） Ａｌｇｏｒｉｔｈｍｓ, ＲＦＣ ５８３０．）の鍵スケジュール部は、４ｎ－ｂｉｔの秘密鍵Ｋを（ｎ／２）－ｂｉｔ毎、８つに等分割し、それぞれを順番にラウンド鍵とする構造になっている

[0057] これは、秘密鍵Ｋに対する一切の演算を行わず、ラウンド毎のセレクタの

みで鍵スケジュール部を構成できるため、ハードウェア実装の際に必要な回路が非常に少ないという特長を持つ。

[0058] しかしながら、図１４に示すように、ＧＯＳＴのデータ構造はＦｅｉｓｔｅｌ構造を取っており、ラウンド鍵を各Ｆ関数の前に排他的論理和しているため、関連鍵攻撃に対して脆弱であることが知られており、実際に確率１で攻撃に成功してしまうことが知られている。

[0059] （ＧＯＳＴへの確率１で成功する関連鍵攻撃）

秘密鍵Ｋ０から得られるラウンド鍵を$RK0_1$，$RK0_2$，…，$RK0_R$とし、秘密鍵Ｋ１から得られるラウンド鍵を$RK1_1$，$RK1_2$，…，$RK1_R$とする。このとき、秘密鍵Ｋ０と秘密鍵Ｋ１の差分を秘密鍵差分Δとし、各ラウンド鍵の差分をラウンド鍵差分$\Delta_1$，$\Delta_2$，…，$\Delta_R$とする。

[0060] ここで、図１５に示すように、任意の秘密鍵差分Δを与えることのできる攻撃者を想定する。このとき、鍵スケジュール部に複雑な非線形関数を導入しているＣＬＥＦＩＡ，Ｃａｍｅｌｌｉａなどにおいては、各ラウンド鍵差分$\Delta_i$は一意には定まらず、データ暗号化部への攻撃が困難となる。

[0061] しかし、図１６に示すように、置換型鍵スケジュール部を持つものでは、各ラウンド鍵差分$\Delta_i$は秘密鍵差分Δをｍ－ｂｉｔ毎に分割したものとなる。例えばＧＯＳＴにおいて，Δ＝（ｄ，ｄ，ｄ，ｄ，ｄ，ｄ，ｄ，ｄ）とした場合（Δは４ｎ－ｂｉｔ，ｄは（ｎ／２）－ｂｉｔであり，ＧＯＳＴの場合はｎ＝６４のため，それぞれ２５６－ｂｉｔ，３２－ｂｉｔ），全てのラウンド鍵差分$\Delta_i$はｄと等しくなる。

[0062] このとき、Ｆｅｉｓｔｅｌ構造を取り、各ラウンド（Ｒ１，Ｒ２，Ｒ３・・・）に提供するラウンド鍵を各Ｆ関数の前に排他的論理和している場合、図１７に示すように、平文差分として（ｄ，ｄ）を与えることでＦ関数への入力差分が０となる。入力差分が０であるＦ関数の出力差分はかならず０になるため，次のラウンド関数への入力差分は（ｄ，ｄ）となる。

同様に全てのラウンド関数において，（ｄ，ｄ）の差分が伝播していくため，暗号文差分が必ず（ｄ，ｄ）となる。

[0063] このため、ＧＯＳＴに秘密鍵Ｋ，平文Ｐを入力したときに得られる暗号文をＣとすると、

秘密鍵Ｋ（＋）（ｄ，ｄ，ｄ，ｄ，ｄ，ｄ，ｄ，ｄ），

平文Ｐ（＋）（ｄ，ｄ）、

を入力したときに得られる暗号文Ｃ’は確率１でＣ’＝Ｃ（＋）（ｄ，ｄ）となる。

[0064] このような等式が成り立つ確率を、関連鍵攻撃を用いた識別攻撃の成功確率と呼ぶ。本来、関連鍵攻撃に対し強いとされるブロック暗号はこの確率が十分に小さいことを示す必要がある。

[0065] また、例えば図１８に示すようなｔｙｐｅ－２一般化Ｆｅｉｓｔｅｌ構造と呼ばれる構造においても，置換型鍵スケジュール部を用いて、Ｆ関数の前にラウンド鍵を排他的論理和する場合、同様に確率１で遷移する関連鍵攻撃を構成可能である。

[0066] その他の置換型鍵スケジュール部を持つ暗号として、ＫＡＳＵＭＩ（非特許文献６：３ｒｄ Ｇｅｎｅｒａｔｉｏｎ Ｐａｒｔｎｅｒｓｈｉｐ Ｐｒｏｊｅｃｔ，Ｔｅｃｈｎｉｃａｌ Ｓｐｅｃｉｆｉｃａｔｉｏｎ Ｇｒｏｕｐ Ｓｅｒｖｉｃｅｓ ａｎｄ Ｓｙｓｔｅｍ Ａｓｐｅｃｔｓ，３Ｇ Ｓｅｃｕｒｉｔｙ，Ｓｐｅｃｉｆｉｃａｔｉｏｎ ｏｆ ｔｈｅ ３Ｇ ＰＰ Ｃｏｎｆｉｄｅｎｔｉａｌｉｔｙ ａｎｄ Ｉｎｔｅｇｒｉｔｙ Ａｌｇｏｒｉｔｈｍｓ；Ｄｏｃｕｍｅｎｔ ２：ＫＡＳＵＭＩ Ｓｐｅｃｉｆｉｃａｔｉｏｎ，Ｖ９．０．０，２００９）が挙げられる。

しかしながら、ＫＡＳＵＭＩの鍵スケジュール部も高い実装性能を持つものの、関連鍵攻撃に対して脆弱であることが知られている。

[0067] ［４．関連鍵攻撃に対し安全性を得られる置換型鍵スケジュール部について］

以上のような問題点を鑑み，以下では、実装コストを増大させずに，関連鍵攻撃に対し安全性を得られるような置換型鍵スケジュール部の構成法について説明する。

[0068] 本方式はまず秘密鍵Ｋを複数個に等分し、それぞれを以下の手法にしたがってデータ暗号化部に供給するものとする。

（手法１）ラウンド鍵挿入位置の変更：ラウンド鍵を従来のＦ関数の前（例えば図１４，図１８に示す位置）ではなく、Ｆ関数の後で排他的論理和を取る（例えば図１９，図２０に示す位置）

（手法２）ラウンド鍵生成置換の変更：データ暗号化部の構造にしたがって，関連鍵攻撃等に対して安全になるような置換法を用いてラウンド鍵を生成する。

[0069] このように鍵スケジュール部を構成することで以下のような効果が得られる。

（効果１）高い実装性能

置換型鍵スケジュール部を用いることで従来型同様高い実装性能を持つことが期待される．特にハードウェア実装時の必要回路が少なくできるという利点がある。

（効果２）ラウンド鍵の挿入位置の変更による安全性向上

図１９や図２０に示す構成のように、ラウンド鍵をＦ関数の前でなく後に排他的論理和することで、置換型鍵スケジュール部を用いたとしても、従来法であるＧＯＳＴのような確率１で成功する関連鍵攻撃を防ぐことが可能となる。

また、実装手法によっては、データ暗号化部の実装性能をさらに向上させることにもつなげられる。

[0070] （効果３）ラウンド鍵の置換法の変更による安全性向上

従来方式と異なり、分割した秘密鍵を順番に供給するのではなく、関連鍵攻撃耐性を持つように適切に順序を入れ替えて供給する。

この順序は秘密鍵Ｋのｂｉｔ長と分割数、及び、データ暗号化部の構造に関連があり、データ暗号化部の構造毎に鍵スケジュール部を設計する必要がある。

[0071] 図１９に示す１ラウンドに１つのＦ関数を実行するＦｅｉｓｔｅｌ構造に

おいては、１つのラウンド鍵の長さは（ｎ／２）－ｂｉｔであり、秘密鍵Ｋが２ｎ－ｂｉｔの場合には（ｎ／２）－ｂｉｔ毎に４等分し、４ラウンド毎に入れ替えながら供給する。

図２１に示す設定である。

[0072] 図２１に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、２ｎ－ｂｉｔの鍵データである。

この２ｎ－ｂｉｔ秘密鍵Ｋを４等分して４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４を生成する。

４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４は、（ｎ／２）－ｂｉｔの鍵データである。

この４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４を、４ラウンドを１単位として、各単位で異なるシーケンスで利用する。

[0073] 図２１に示すように、最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケンスでラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１

ラウンドＲ２：ラウンド鍵Ｋ２

ラウンドＲ３：ラウンド鍵Ｋ３

ラウンドＲ４：ラウンド鍵Ｋ４

[0074] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスでラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ３

ラウンドＲ６：ラウンド鍵Ｋ１

ラウンドＲ７：ラウンド鍵Ｋ４

ラウンドＲ８：ラウンド鍵Ｋ２

[0075] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスでラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ４

ラウンドＲ１０：ラウンド鍵Ｋ３

ラウンドＲ１１：ラウンド鍵Ｋ２

ラウンドＲ１２：ラウンド鍵Ｋ１

このように、４ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用する。

[0076] また、ラウンド鍵の長さよりも小さい単位で分割することも可能であり、２ｎ－ｂｉｔの秘密鍵Ｋを（ｎ／４）－ｂｉｔ毎に８等分し、それらを４ラウンド毎に入れ替えながら、（ｎ／４）－ｂｉｔのものを２つずつ供給していくことも可能である。

例えば、図２２に示す構成である。

[0077] 図２２に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、２ｎ－ｂｉｔの鍵データである。

この２ｎ－ｂｉｔ秘密鍵Ｋを８等分して８つの鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５，Ｋ６，Ｋ７，Ｋ８を生成する。

８つの鍵Ｋ１～Ｋ８は、（ｎ／４）－ｂｉｔの鍵データである。

この８つの鍵Ｋ１～Ｋ８から選択した２つの鍵の連結データからなる（ｎ／２）－ｂｉｔラウンド鍵：ＫｘＫｙを、４ラウンドを１単位として、各単位で異なるシーケンスで利用する。

[0078] 図２２に示すように、最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケンスで鍵データＫ１～Ｋ８を連結したラウンド鍵ＫｘＫｙを入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５Ｋ６

ラウンドＲ４：ラウンド鍵Ｋ７Ｋ８

[0079] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスで鍵データＫ１～Ｋ８を連結したラウンド鍵ＫｘＫｙを入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ３Ｋ４

ラウンドＲ６：ラウンド鍵Ｋ２Ｋ７

ラウンドＲ７：ラウンド鍵Ｋ１Ｋ６

ラウンドＲ８：ラウンド鍵Ｋ５Ｋ８

[0080] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスで鍵データＫ１～Ｋ８を連結したラウンド鍵ＫｘＫｙを入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ２Ｋ７

ラウンドＲ１０：ラウンド鍵Ｋ４Ｋ５

ラウンドＲ１１：ラウンド鍵Ｋ３Ｋ６

ラウンドＲ１２：ラウンド鍵Ｋ１Ｋ８

このように、４ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用する。

[0081] また、秘密鍵Ｋの長さが４ｎ－ｂｉｔの場合には（ｎ／２）－ｂｉｔ毎に８等分し、８ラウンド毎に入れ替えながら供給していく。

例えば図２３に示す構成である。

[0082] 図２３に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、４ｎ－ｂｉｔの鍵データである。

この４ｎ－ｂｉｔ秘密鍵Ｋを８等分して８つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５，Ｋ６，Ｋ７，Ｋ８を生成する。

８つのラウンド鍵Ｋ１～Ｋ８は、（ｎ／２）－ｂｉｔの鍵データである。

この８つのラウンド鍵Ｋ１～Ｋ８を、８ラウンドを１単位として、各単位で異なるシーケンスで利用する。

[0083] 図２３に示すように、最初の８ラウンド：Ｒ１～Ｒ８では以下のシーケンスでラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１

ラウンドＲ２：ラウンド鍵Ｋ２

ラウンドＲ３：ラウンド鍵Ｋ３

ラウンドＲ４：ラウンド鍵Ｋ４

ラウンドＲ５：ラウンド鍵Ｋ５

ラウンドＲ６：ラウンド鍵Ｋ６

ラウンドＲ７：ラウンド鍵Ｋ７

ラウンドＲ８：ラウンド鍵Ｋ８

[0084] 次の８ラウンド：Ｒ９～Ｒ１６では以下のシーケンスでラウンド鍵を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ２

ラウンドＲ１０：ラウンド鍵Ｋ５

ラウンドＲ１１：ラウンド鍵Ｋ１

ラウンドＲ１２：ラウンド鍵Ｋ４

ラウンドＲ１３：ラウンド鍵Ｋ８

ラウンドＲ１４：ラウンド鍵Ｋ６

ラウンドＲ１５：ラウンド鍵Ｋ３

ラウンドＲ１６：ラウンド鍵Ｋ７

このように、８ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用する。

[0085] また、図２０に示すような、１ラウンドに２つのＦ関数を同時実行する４系列一般化Ｆｅｉｓｔｅｌ構造において、ラウンド鍵は（ｎ／４）－ｂｉｔのものが２つとなる。秘密鍵Ｋが２ｎ－ｂｉｔの場合には（ｎ／４）－ｂｉｔ毎に８等分し、４ラウンド毎に入れ替えながら（ｎ／４）－ｂｉｔのものを２つずつ供給していく。

例えば図２４に示す構成である。

[0086] 図２４に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、２ｎ－ｂｉｔの鍵データである。

この２ｎ－ｂｉｔ秘密鍵Ｋを８等分して８つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５，Ｋ６，Ｋ７，Ｋ８を生成する。

８つのラウンド鍵Ｋ１～Ｋ８は、（ｎ／４）－ｂｉｔの鍵データである。

この８つのラウンド鍵Ｋ１～Ｋ８から選択した２つのラウンド鍵を、４ラウンドを１単位として、各単位で異なるシーケンスで利用する。

[0087] 図２４に示すように、最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケン

スで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ６

ラウンドＲ４：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ８

[0088] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ１

ラウンドＲ６：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ６

ラウンドＲ７：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ３

ラウンドＲ８：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ８

[0089] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ５

ラウンドＲ１０：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ３

ラウンドＲ１１：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ２

ラウンドＲ１２：ラウンド鍵Ｋ６と、ラウンド鍵Ｋ８

このように、４ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用する。

[0090] また、実装効率をさらに高めた手法として、（ｎ／４）－ｂｉｔ毎８等分した中で、４つを左側のＦ関数へのラウンド鍵$ＲＫ_{r,0}$に用いるものとし、残りの４つを右側のＦ関数へのラウンド鍵$ＲＫ_{r,1}$に用いるものとする。

例えば図２５に示す構成である。

[0091] 図２５に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、２ｎ－ｂｉｔの鍵データである。

この２ｎ－ｂｉｔ秘密鍵Ｋを８等分して８つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５，Ｋ６，Ｋ７，Ｋ８を生成する。

８つのラウンド鍵Ｋ１～Ｋ８は、（ｎ／４）－ｂｉｔの鍵データである。

この８つのラウンド鍵Ｋ１～Ｋ８から選択した２つのラウンド鍵を、４ラウンドを１単位として、各単位で異なるシーケンスで利用する。

この例では、Ｋ１～Ｋ４のラウンド鍵を４系列Ｆｅｉｓｔｅｌ構造における各ラウンドの左側のＦ関数に適用し、Ｋ５～Ｋ８のラウンド鍵を４系列Ｆｅｉｓｔｅｌ構造における各ラウンドの右側のＦ関数に適用する。

[0092] 図２５に示すように、最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ５

ラウンドＲ２：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ６

ラウンドＲ３：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ７

ラウンドＲ４：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ８

[0093] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ５

ラウンドＲ６：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ８

ラウンドＲ７：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ６

ラウンドＲ８：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ７

[0094] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ５

ラウンドＲ１０：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ７

ラウンドＲ１１：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ８

ラウンドＲ１２：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ６

このように、４ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用する。

[0095] この図２５に示す構成では、ラウンド鍵Ｋ１～Ｋ８の８つの鍵の入力順を４ラウンド単位で全て入れ替えるのではなく、４つ毎、すなわち、

ラウンド鍵：Ｋ１～Ｋ４、

ラウンド鍵：Ｋ５～Ｋ８、

これらの４つのラウンド鍵単位で入力順の入れ替えを行う。

この構成によって、置換型鍵スケジュール部に必要となるセレクタのコストをさらに削減することが可能になる。

[0096] また、例えば、図２６に示すような４系列一般化Ｆｅｉｓｔｅｌ構造におけるｃｙｃｌｉｃ ｓｈｉｆｔをｒｏｕｎｄ ｐｅｒｍｕｔａｔｉｏｎに変更したモデル（４系列一般化Ｆｅｉｓｔｅｌ＋とする）に対しても，本方式による鍵スケジュール部が有効である。

[0097] 図２６に示すような４系列一般化Ｆｅｉｓｔｅｌ構造におけるｃｙｃｌｉｃ ｓｈｉｆｔをｒｏｕｎｄ ｐｅｒｍｕｔａｔｉｏｎに変更したモデル（４系列一般化Ｆｅｉｓｔｅｌ＋）につい説明する。

[0098] 図２６に示す構成は、ｄ－ｌｉｎｅの一般化Ｆｅｉｓｔｅｌ構造において、ｎビット入力データをｎ／ｄビット毎ｄ個に分割しそれぞれＦ関数処理，排他的論理和処理を行う構成を基本構成とし、ラウンド鍵との演算は、Ｆ関数の出力に対して実行する構成である。

このとき、

Ｆ関数に入力されるデータ系列をＦ関数入力側データ系列，

排他的論理和されるデータ系列を排他的論理和側データ系列、

とする。

各系列（各ライン）において転送されるｎ／ｄビットデータ各々について、さらにｄ／２個に再分割する（この際の分割は等分割でなくても良い）。

各系列（各ライン）各々でｄ／２個に再分割されたデータを次のルールにしたがって分配する。

[0099] （１）Ｆ関数入力側データ系列は必ず次のラウンド関数の排他的論理和側データ系列に分配する

（２）排他的論理和側データ系列は必ず次のラウンド関数のＦ関数入力側データ系列に分配する

（３）各ｄ／２個に分割されたデータ系列はｄ／２箇所の次のラウンド関

数のデータ系列にそれぞれ重複なく分配する

[0100] このように分配した後、各ｄ／２個に分割されたデータをそれぞれ１つのデータに結合する。

これを必要回数繰り返す。

[0101] このｃｙｃｌｉｃ ｓｈｉｆｔをｒｏｕｎｄ ｐｅｒｍｕｔａｔｉｏｎに変更したモデル（４系列一般化Ｆｅｉｓｔｅｌ＋）に対しても，本方式による鍵スケジュール部が有効である。

具体的には、例えば、図２５と同様の入れ替え手法である図２７に示す構造が適している。

すなわち、秘密鍵を（ｎ／４）－ｂｉｔ毎８等分した中で、４つを左側のＦ関数へのラウンド鍵$RK_{r,0}$に用いるものとし、残りの４つを右側のＦ関数へのラウンド鍵$RK_{r,1}$に用いる構成である。

[0102] 図２７に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、２ｎ－ｂｉｔの鍵データである。

この２ｎ－ｂｉｔ秘密鍵Ｋを８等分して８つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５，Ｋ６，Ｋ７，Ｋ８を生成する。

８つのラウンド鍵Ｋ１～Ｋ８は、（ｎ／４）－ｂｉｔの鍵データである。

この８つのラウンド鍵Ｋ１～Ｋ８から選択した２つのラウンド鍵を、４ラウンドを１単位として、各単位で異なるシーケンスで利用する。

この例では、Ｋ１～Ｋ８のラウンド鍵から選択した２つのラウンド鍵を、ｃｙｃｌｉｃ ｓｈｉｆｔをｒｏｕｎｄ ｐｅｒｍｕｔａｔｉｏｎに変更した４系列一般化Ｆｅｉｓｔｅｌ＋構造における各ラウンドの２つのＦ関数に適用する。

[0103] 図２７に示すように、最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ５

ラウンドＲ２：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ６

ラウンドＲ３：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ７

ラウンドＲ４：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ８

[0104] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ５

ラウンドＲ６：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ８

ラウンドＲ７：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ６

ラウンドＲ８：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ７

[0105] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ５

ラウンドＲ１０：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ７

ラウンドＲ１１：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ８

ラウンドＲ１２：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ６

このように、４ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用する。

[0106] 次に、秘密鍵のｂｉｔ長が１ラウンドに必要なラウンド鍵のｂｉｔ長の整数倍でない場合の置換型鍵スケジュール部の構成例を示す。

例えば、秘密鍵が（５ｎ／４）－ｂｉｔであり、１ラウンドに必要なラウンド鍵が（ｎ／２）－ｂｉｔのときの鍵スケジュール部の構成法を図２８に示す。

この構成では、初めに１ラウンドに必要なラウンド鍵のｂｉｔ長の整数倍になるように置換と拡張を行う。その後は順番に供給していく構造を取る。

[0107] 図２８に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、（５／４）ｎ－ｂｉｔの鍵データである。

まず、置換型鍵スケジュール部は、この（５／４）ｎ－ｂｉｔの鍵データに基づいて、ラウンド鍵のｂｉｔ長に相当する５つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５を生成する。

このラウンド鍵Ｋ１～Ｋ５から選択した２つのラウンド鍵を各ラウンドに

適用するラウンド鍵として入力して２つのＦ関数に適用する。

[0108] 図２８に示すように、最初の５ラウンド：Ｒ１～Ｒ５では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ２：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ３：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ４：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ５

ラウンドＲ５：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

次の５ラウンド：Ｒ６～Ｒ１０、その後の５ラウンド：Ｒ１１～Ｒ１５でも同様のシーケンスで２つのラウンド鍵を入力して適用する。

[0109] 上述の「（効果３）ラウンド鍵の置換法の変更による安全性向上」の効果について詳細に説明を行う。

まず、差分確率，ａｃｔｉｖｅ　Ｆ関数、最小ａｃｔｉｖｅ　Ｆ関数数の定義を行う。

[0110] 差分攻撃は、ある入力差分からある出力差分に高い確率で伝播するようなものを利用して攻撃する。つまり、安全性を考える際にはどのような入力差分、出力差分の組を考えても、高い確率で伝播するようなものが存在しないことを示す必要がある。

[0111] 関連鍵差分攻撃も同様に、ある入力差分とある秘密鍵差分から、ある出力差分に高い確率で伝播するようなものを利用して攻撃する。つまり、安全性を考える際にはどのような入力差分、秘密鍵差分、出力差分の組を考えても高い確率で伝播するようなものが存在しないことを示す必要がある。このような、ある入力差分がある出力差分に伝播する確率、及び、ある入力差分とある秘密鍵差分がある出力差分に伝播する確率を差分確率と定義する。前述した通り、ＧＯＳＴではこの差分確率が１となるものが存在する。

[0112] このような差分確率は、非ゼロの入力差分が与えられた非線形関数（Ｆ関数）でのみ低下することが知られている。この非ゼロの入力差分が与えられた非線形関数（Ｆ関数）をａｃｔｉｖｅ　Ｆ関数と定義する。ａｃｔｉｖｅ

Ｆ関数の個数は差分攻撃に対する安全性と密接に関係しており、ある入力差分に対して、多くのａｃｔｉｖｅ　Ｆ関数が得られるのであれば、十分に安全だと考えることができる。

[0113] ａｃｔｉｖｅ　Ｆ関数の数は、入力差分を一つ定めれば決めることができる。以上より、差分攻撃に対する安全性を考える際にはどのような入力差分を与えたとしても十分に多くのａｃｔｉｖｅ　Ｆ関数が得られるということを示せばよいということが分かる。このような各入力差分に対するａｃｔｉｖｅ　Ｆ関数の個数の最小値を最小ａｃｔｉｖｅ　Ｆ関数数と定義する。

[0114] 例えば図２０に示すような、ラウンド鍵をＦ関数の出力側に排他的論理和を取る４系列一般化Ｆｅｉｓｔｅｌ構造において、図２９に示すようにラウンド鍵を単純に順番に入れる従来法を採用した際の、関連鍵差分を考慮したラウンド数と最小ａｃｔｉｖｅ　Ｆ関数数の対応を以下の（表１）に示す。

[0115]

[表1]

（表1）

| ラウンド数 | 最小active F関数数 | ラウンド数 | 最小active F関数数 |
|---|---|---|---|
| 1 | 0 | 16 | 4 |
| 2 | 0 | 17 | 4 |
| 3 | 0 | 18 | 4 |
| 4 | 0 | 19 | 4 |
| 5 | 0 | 20 | 5 |
| 6 | 1 | 21 | 5 |
| 7 | 1 | 22 | 5 |
| 8 | 2 | 23 | 5 |
| 9 | 2 | 24 | 6 |
| 10 | 2 | 25 | 6 |
| 11 | 2 | 26 | 6 |
| 12 | 3 | 27 | 6 |
| 13 | 3 | 28 | 7 |
| 14 | 3 | 29 | 7 |
| 15 | 3 | 30 | 7 |

[0116] この方式の場合、例えば関連鍵差分攻撃に対する安全性を保証するために必要なａｃｔｉｖｅ　Ｆ関数の個数が７個であるような暗号では、少なくとも２８ラウンドが必要になることが分かる。

[0117] 例えば図２０に示すような４系列一般化Ｆｅｉｓｔｅｌ構造に、図２４に示すような本開示に従った処理、すなわち、置換型鍵スケジュール部において、所定のラウンド単位で、入力鍵シーケンスを変更する処理を行う構成に対して、関連鍵差分を考慮したラウンド数と最小ａｃｔｉｖｅ　Ｆ関数数の対応を（表２）に示す。

[0118] [表2]

（表2）

| ラウンド数 | 最小active F関数数 | ラウンド数 | 最小active F関数数 |
|---|---|---|---|
| 1 | 0 | 16 | 7 |
| 2 | 0 | 17 | 8 |
| 3 | 0 | 18 | 8 |
| 4 | 0 | 19 | 9 |
| 5 | 0 | 20 | 9 |
| 6 | 0 | 21 | 10 |
| 7 | 1 | 22 | 11 |
| 8 | 1 | 23 | 11 |
| 9 | 2 | 24 | 12 |
| 10 | 3 | 25 | 12 |
| 11 | 4 | 26 | 12 |
| 12 | 4 | 27 | 13 |
| 13 | 5 | 28 | 14 |
| 14 | 6 | 29 | 15 |
| 15 | 7 | 30 | 16 |

[0119] 上記の表２では、例えば関連鍵差分攻撃に対する安全性を保証するために必要なａｃｔｉｖｅ　Ｆ関数の個数が７個であるような暗号では、少なくとも１５ラウンドが必要になることが分かる。

先の表１では、関連鍵差分攻撃に対する安全性を保証するために必要なａｃｔｉｖｅ　Ｆ関数の個数が７個であるような暗号では、少なくとも２８ラウンドが必要になる。

本開示に従った処理、すなわち、置換型鍵スケジュール部において、所定

のラウンド単位で、入力鍵シーケンスを変更する処理を行う構成とすることで、従来法と比べて１３ラウンド減らせることが分かる。

[0120] このように、本開示に従った処理、すなわち、置換型鍵スケジュール部において、所定のラウンド単位で、入力鍵シーケンスを変更する処理を行う構成とすることで、このような鍵入れ替えを実行しない従来法より多くのａｃｔｉｖｅ　Ｆ関数の個数を保証できることが分かる。

[0121] 同様に、図２６に示すような、４系列一般化Ｆｅｉｓｔｅｌ構造におけるｃｙｃｌｉｃ　ｓｈｉｆｔをｒｏｕｎｄ　ｐｅｒｍｕｔａｔｉｏｎに変更したモデル４系列一般化Ｆｅｉｓｔｅｌ＋構造について考察する。

この構造に対して、ラウンド鍵を単純に順番に入れる従来法（図３０）を採用した際の、関連鍵差分を考慮したラウンド数と最小ａｃｔｉｖｅ　Ｆ関数数の対応は、前記の（表１）と同じ値を取る。

[0122] この方式の場合、例えば関連鍵差分攻撃に対する安全性を保証するために必要なａｃｔｉｖｅ　Ｆ関数の個数が７個であるような暗号では、少なくとも２８ラウンドが必要になることが分かる。

[0123] また、図２６に示すような、４系列一般化Ｆｅｉｓｔｅｌ＋構造に対して、図２４に示すような本発明による鍵スケジュール部の構成とした場合において、関連鍵差分を考慮した際のラウンド数と最小ａｃｔｉｖｅ　Ｆ関数数の対応を、以下の（表３）に示す。

[0124]

[表3]

（表3）

| ラウンド数 | 最小active F関数数 | ラウンド数 | 最小active F関数数 |
|---|---|---|---|
| 1 | 0 | 16 | 7 |
| 2 | 0 | 17 | 8 |
| 3 | 0 | 18 | 9 |
| 4 | 0 | 19 | 10 |
| 5 | 0 | 20 | 10 |
| 6 | 0 | 21 | 11 |
| 7 | 0 | 22 | 12 |
| 8 | 1 | 23 | 12 |
| 9 | 3 | 24 | 13 |
| 10 | 3 | 25 | 14 |
| 11 | 4 | 26 | 15 |
| 12 | 5 | 27 | 15 |
| 13 | 5 | 28 | 16 |
| 14 | 6 | 29 | 17 |
| 15 | 7 | 30 | 17 |

[0125] 上記の表３では、例えば関連鍵差分攻撃に対する安全性を保証するために必要なａｃｔｉｖｅ　Ｆ関数の個数が７個であるような暗号では、少なくとも１５ラウンドが必要になることが分かる。

先の表１では、関連鍵差分攻撃に対する安全性を保証するために必要なａｃｔｉｖｅ　Ｆ関数の個数が７個であるような暗号では、少なくとも２８ラウンドが必要になる。

本開示に従った処理、すなわち、置換型鍵スケジュール部において、所定のラウンド単位で、入力鍵シーケンスを変更する処理を行う構成とすること

で、従来法と比べて１３ラウンド減らせることが分かる。

[0126] このように、本開示に従った処理、すなわち、置換型鍵スケジュール部において、所定のラウンド単位で、入力鍵シーケンスを変更する処理を行う構成とすることで、このような鍵入れ替えを実行しない従来法より多くのａｃｔｉｖｅ Ｆ関数の個数を保証でき、安全性の高い暗号処理構成を少ないラウンド数で実現することが可能となる。

[0127] ［５．鍵置換型鍵スケジュール部の様々な構成例（バリエーション）について］

次に、鍵置換型鍵スケジュール部の様々な構成例（バリエーション）について説明する。

[0128] 分割数ｄとしたｄ系列一般化Ｆｅｉｓｔｅｌ構造をもつｎ－ｂｉｔブロック暗号においては、図３１に示すように、一般に、１ラウンド辺り、（ｎ／ｄ）－ｂｉｔのラウンド鍵を（ｄ／２）個、利用する。

従って、１ラウンド辺り、合計で、

（ｎ／ｄ）×（ｄ／２）

＝（ｎ／２）－ｂｉｔのラウンド鍵データが必要となる。

[0129] 例えば、先に説明した図１８に示すような４系列一般化Ｆｅｉｓｔｅｌ構造（ｄ＝４）を適用する場合、ラウンド鍵の入力処理としては、

各ラウンド単位で、（ｎ／４）－ｂｉｔのラウンド鍵を２つずつ入力する必要がある。

[0130] ラウンド鍵の生成元となる秘密鍵の長さに応じて以下のようなラウンド鍵の生成、入力が実行される。

秘密鍵がｎ－ｂｉｔの場合には、例えば図３２に示すように、ｎ－ｂｉｔ秘密鍵を４等分したｎ／４－ｂｉｔのラウンド鍵を生成して各ラウンドに２つずつ入力する。

秘密鍵が（５／４）ｎ－ｂｉｔの場合には、例えば図３３、図３４に示すように、ｎ－ｂｉｔ秘密鍵を５等分したｎ／４－ｂｉｔのラウンド鍵を生成して各ラウンドに２つずつ入力する。

[0131] 例えば、図３２に示すように秘密鍵ｎ－ｂｉｔを４等分した４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４を利用する構成では、２ラウンド毎に鍵置換を行う、すなわち、２ラウンド毎に鍵供給シーケンスを変更する。

図３２に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、ｎ－ｂｉｔの鍵データである。

このｎ－ｂｉｔ秘密鍵Ｋを４等分して４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４を生成する。

４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４は、（ｎ／４）－ｂｉｔの鍵データである。

この４つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４を、４ラウンドを１単位として、各単位で異なるシーケンスで利用する。

[0132] 図２１に示すように、最初の２ラウンド：Ｒ１～Ｒ２では以下のシーケンスで２つのラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

[0133] 次の２ラウンド：Ｒ３～Ｒ４では以下のシーケンスでラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ３：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ１

ラウンドＲ４：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ２

[0134] 次の２ラウンド：Ｒ５～Ｒ６では以下のシーケンスでラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ３

ラウンドＲ６：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ１

[0135] 次の２ラウンド：Ｒ７～Ｒ８では以下のシーケンスでラウンド鍵Ｋ１～Ｋ４を入力して適用する。

ラウンドＲ７：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ４

ラウンドＲ８：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ３

[0136] 次の２ラウンド：Ｒ９～Ｒ１０では以下のシーケンスでラウンド鍵Ｋ１～

Ｋ４を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ１０：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

[0137] また、図３３に示すように秘密鍵（５／４）ｎ－ｂｉｔを５等分した５つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５を利用する構成では、これらの５つの鍵Ｋ１～Ｋ５の、５つの鍵選択後に鍵置換を行うため、５ラウンドに２回の鍵置換を行う。すなわち、５ラウンド毎に２回、鍵供給シーケンスを変更する。

[0138] 図３３に示すラウンド鍵の供給処理について説明する。

秘密鍵Ｋは、（５／４）ｎ－ｂｉｔの鍵データである。

まず、置換型鍵スケジュール部は、この（５／４）ｎ－ｂｉｔの鍵データに基づいて、ラウンド鍵のｂｉｔ長に相当する５つのラウンド鍵Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５を生成する。

このラウンド鍵Ｋ１～Ｋ５から選択した２つのラウンド鍵を各ラウンドに適用するラウンド鍵として入力して２つのＦ関数に適用する。

[0139] 図３３に示すように、最初の５ラウンド：Ｒ１～Ｒ５では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ５

ラウンドＲ４：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ３

ラウンドＲ５：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

次の５ラウンド：Ｒ６～Ｒ１０でも同様のシーケンスで２つのラウンド鍵を入力して適用する。

[0140] なお、この図３３に示す鍵供給処理構成における鍵の供給シーケンスは、結果として、先に説明した図２８と同様となる。

すなわち、いずれも、Ｋ１，Ｋ２，Ｋ３，Ｋ４，Ｋ５，Ｋ５，Ｋ５，Ｋ１，Ｋ２，Ｋ３・・・この順番で鍵の供給が行われる。

このように、置換型鍵スケジュール部の鍵供給処理において、例えばｍラウンドにｍ’（＞１）回の鍵置換が必要な設定では、２回目以降の鍵置換を行わず、鍵の選択順序を変更する処理を行う構成としてもよい。

例えば、図３３に示す鍵供給処理シーケンスと、図３４に示す鍵供給処理シーケンスは、全く同じとなる。

すなわち、最初の５ラウンド：Ｒ１～Ｒ５では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ５

ラウンドＲ４：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ３

ラウンドＲ５：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

次の５ラウンド：Ｒ６～Ｒ１０でも同様のシーケンスで２つのラウンド鍵を入力して適用する。

[0141] また、鍵置換処理として、分割した鍵の供給順番を並び替えて、順にラウンド関数に入力していくのではなく、複数ラウンド単位で鍵の選択順序を変更することでも全く同様の効果が得られる。例えば、先に図２４を参照して説明したラウンド鍵供給構成と、図３５に示す鍵供給処理シーケンスは、全く同じとなる。

すなわち、最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ６

ラウンドＲ４：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ８

[0142] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ１

ラウンドＲ６：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ６

ラウンドＲ７：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ３

ラウンドＲ８：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ８

[0143] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ５

ラウンドＲ１０：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ３

ラウンドＲ１１：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ２

ラウンドＲ１２：ラウンド鍵Ｋ６と、ラウンド鍵Ｋ８

このように、図２４を参照して説明したラウンド鍵供給構成と、図３５に示す鍵供給処理シーケンスは、４ラウンドを１単位として、各単位で異なるラウンド鍵入力シーケンスを適用した構成となる。

[0144] （複数種類の鍵置換）

先に図２４を参照して説明した構成では、鍵の長さが２ｎ－ｂｉｔで８等分する場合、４ラウンド毎に同じ態様の鍵置換を行っている。

これに対して、例えば、図３６に示すように、４ラウンド単位で実行する置換処理を毎回異なる態様に設定してもよい。

[0145] 図３６に示す構成による鍵供給シーケンスは以下の通りである。最初の４ラウンド：Ｒ１～Ｒ４では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ６

ラウンドＲ４：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ８

[0146] 次の４ラウンド：Ｒ５～Ｒ８では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ５：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ１

ラウンドＲ６：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ６

ラウンドＲ７：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ３

ラウンドＲ８：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ８

[0147] 次の４ラウンド：Ｒ９～Ｒ１２では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ９　：ラウンド鍵Ｋ８と、ラウンド鍵Ｋ５

ラウンドＲ１０：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ３

ラウンドＲ１１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ１２：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ６

[0148] この図３６に示す鍵供給シーケンスは、ラウンド１～４，５～８までは、図２４に示す鍵供給シーケンスと同じであるが、ラウンド９以降は異なるシーケンスである。

図２４に示す設定では置換型鍵スケジュール部の実行する鍵置換、すなわち鍵入れ替え処理を４ラウンド単位で同じ設定で実行する構成であるが、図３６に示す構成では、置換型鍵スケジュール部の実行する鍵置換、すなわち鍵入れ替え処理を４ラウンド単位で異なる設定で実行しているからである。

[0149] また、分割した鍵の順番を並び替えて、順にラウンド関数に入力していくのではなく、鍵の選択順序を変更することでも全く同様の効果が得られる。例えば、先に図３６を参照して説明したラウンド鍵供給構成と、図３７に示す鍵供給処理シーケンスは、全く同じとなる。

[0150] （鍵の部分選択）

上記のように、図２４や、図３６に示す構成では、４ラウンドに８等分した８個の鍵を１つずつ入力している。

図３８に示すように、３ラウンド毎に置換を行い３ラウンドに６個のみを利用する構成も可能である。

[0151] 図３８に示す構成による鍵供給シーケンスは以下の通りである。最初の３ラウンド：Ｒ１～Ｒ３では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１：ラウンド鍵Ｋ１と、ラウンド鍵Ｋ２

ラウンドＲ２：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ４

ラウンドＲ３：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ６

[0152] 次の３ラウンド：Ｒ４～Ｒ６では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ４：ラウンド鍵Ｋ７と、ラウンド鍵Ｋ３

ラウンドＲ５：ラウンド鍵Ｋ８と、ラウンド鍵Ｋ１

ラウンドＲ６：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ４

[0153] 次の３ラウンド：Ｒ７～Ｒ９では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ７：ラウンド鍵Ｋ６と、ラウンド鍵Ｋ８

ラウンドＲ８：ラウンド鍵Ｋ５と、ラウンド鍵Ｋ７

ラウンドＲ９：ラウンド鍵Ｋ３と、ラウンド鍵Ｋ１

[0154] 次の３ラウンド：Ｒ１０～Ｒ１２では以下のシーケンスで２つのラウンド鍵を入力して適用する。

ラウンドＲ１０：ラウンド鍵Ｋ４と、ラウンド鍵Ｋ５

ラウンドＲ１１：ラウンド鍵Ｋ２と、ラウンド鍵Ｋ６

ラウンドＲ１２：ラウンド鍵Ｋ８と、ラウンド鍵Ｋ７

この例では、３ラウンド単位で、同一パターンの鍵置換を繰り返す構成としている。

[0155] また、鍵の順番を並び替えて、順にラウンド関数に入力していくのではなく、鍵の選択順序を変更することでも全く同様の効果が得られる。例えば、図３８を参照して説明したラウンド鍵供給構成と、図３９に示す鍵供給処理シーケンスは、全く同じとなる。

[0156] （実装効率）

本開示のラウンド鍵の供給方式は従来方式と同様、置換型鍵スケジュール部による処理として実行するものであるため、実装効率が高い。

例えば、図４０に示すように、ラウンド鍵の生成元となる秘密鍵Ｋの長さが２ｎ－ｂｉｔのとき、ラウンド毎に鍵を８等分した鍵Ｋ１～Ｋ８のうちの

２つのラウンド鍵を選択する。

[0157] ここで、先に説明した図２７のように鍵Ｋ１～Ｋ８を２つに区分して、

Ｋ１，Ｋ２，Ｋ３，Ｋ４

Ｋ５，Ｋ６，Ｋ７，Ｋ８

これらの４つのラウンド鍵のグループを２つ設定し、

これらの各グループ単位で、所定ラウンド数単位の鍵置換、すなわち鍵供給シーケンスの変更を行う構成とすることで、ある実装形態において実装効率をさらに高めることができる。

[0158] 具体的には、図４１に示すように、置換型鍵スケジュール部を２つのセレクタを有する構成にして、各セレクタにおいて、上記の４つのラウンド鍵のグループ、すなわち、

Ｋ１，Ｋ２，Ｋ３，Ｋ４

Ｋ５，Ｋ６，Ｋ７，Ｋ８

これらの各ラウンド鍵のグループからの出力鍵の選択を実行する構成とする。

[0159] 先に説明した図２４のような鍵スケジュール部を設計すると図４０のように８つから１つを選択するセレクタが２つ必要となるが、例えば、図２７のように置換型鍵スケジュール部を構成することで図４１のように４つから１つを選択する小型のセレクタが２つとなる。

[0160] 同様に、（５／４）ｎ－ｂｉｔ鍵を５等分したものを５つ全て置換するのではなく、入力数が３のセレクタを利用できるように置換を設計することで実装効率が上がる。

具体的には、図４２に示すように、

（５／４）ｎビットの秘密鍵Ｋから、ｎ／４ビットのラウンド鍵Ｋ１，Ｋ２，Ｋ，Ｋ４，Ｋ５を生成し、

第１のセレクタが、Ｋ１，Ｋ２，Ｋ３を対象とした鍵選択を実行し、第２のセレクタが、Ｋ３，Ｋ４，Ｋ５を対象とした鍵選択を実行する構成である。

[0161] ［６．暗号処理装置の構成例について］

最後に、上述した実施例に従った暗号処理を実行する暗号処理装置の実相例について説明する。

上述した実施例に従った暗号処理を実行する暗号処理装置は、暗号処理を実行する様々な情報処理装置に搭載可能である。具体的には、ＰＣ、ＴＶ、レコーダ、プレーヤ、通信機器、さらに、ＲＦＩＤ、スマートカード、センサネットワーク機器、デンチ／バッテリー認証モジュール、健康、医療機器、自立型ネットワーク機器等、例えばデータ処理や通信処理に伴う暗号処理を実行する様々な危機において利用可能である。

[0162] 本開示の暗号処理を実行する装置の一例としてのＩＣモジュール７００の構成例を図４３に示す。上述の処理は、例えばＰＣ、ＩＣカード、リーダライタ、その他、様々な情報処理装置において実行可能であり、図４３に示すＩＣモジュール７００は、これら様々な機器に構成することが可能である。

[0163] 図４３に示すＣＰＵ（Ｃｅｎｔｒａｌ ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ）７０１は、暗号処理の開始や、終了、データの送受信の制御、各構成部間のデータ転送制御、その他の各種プログラムを実行するプロセッサである。メモリ７０２は、ＣＰＵ７０１が実行するプログラム、あるいは演算パラメータなどの固定データを格納するＲＯＭ（Ｒｅａｄ－Ｏｎｌｙ－Ｍｅｍｏｒｙ）、ＣＰＵ７０１の処理において実行されるプログラム、およびプログラム処理において適宜変化するパラメータの格納エリア、ワーク領域として使用されるＲＡＭ（Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）等からなる。また、メモリ７０２は暗号処理に必要な鍵データや、暗号処理において適用する変換テーブル（置換表）や変換行列に適用するデータ等の格納領域として使用可能である。なおデータ格納領域は、耐タンパ構造を持つメモリとして構成されることが好ましい。

[0164] 暗号処理部７０３は、上記において説明した暗号処理構成、すなわち、例えば一般化Ｆｅｉｓｔｅｌ構造や、Ｆｅｉｓｔｅｌ構造を適用した共通鍵ブロック暗号処理アルゴリズムに従った暗号処理、復号処理を実行する。

[0165] なお、ここでは、暗号処理手段を個別モジュールとした例を示したが、このような独立した暗号処理モジュールを設けず、例えば暗号処理プログラムをＲＯＭに格納し、ＣＰＵ７０１がＲＯＭ格納プログラムを読み出して実行するように構成してもよい。

[0166] 乱数発生器７０４は、暗号処理に必要となる鍵の生成などにおいて必要となる乱数の発生処理を実行する。

[0167] 送受信部７０５は、外部とのデータ通信を実行するデータ通信処理部であり、例えばリーダライタ等、ＩＣモジュールとのデータ通信を実行し、ＩＣモジュール内で生成した暗号文の出力、あるいは外部のリーダライタ等の機器からのデータ入力などを実行する。

[0168] なお、上述した実施例において説明した暗号処理装置は、入力データとしての平文を暗号化する暗号化処理に適用可能であるのみならず、入力データとしての暗号文を平文に復元する復号処理にも適用可能である。

暗号化処理、復号処理、双方の処理において、上述した実施例において説明した構成を適用することが可能である。

[0169] ［７．本開示の構成のまとめ］

以上、特定の実施例を参照しながら、本開示の実施例について詳解してきた。しかしながら、本開示の要旨を逸脱しない範囲で当業者が実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本開示の要旨を判断するためには、特許請求の範囲の欄を参酌すべきである。

[0170] なお、本明細書において開示した技術は、以下のような構成をとることができる。

（１） データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、

前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する暗号処理装置。

[0171] （２）前記鍵スケジュール部は、前記ラウンド演算実行部に対する前記複数のラウンド鍵の入力シーケンスを、前記ラウンド演算実行部における複数ラウンド単位で変更する前記（１）に記載の暗号処理装置。

（３）前記暗号処理部は、前記複数のラインに分割されたデータを入力して非線形変換処理と、線形変換処理を含むＦ関数実行部と、前記Ｆ関数実行部の出力に対して、前記ラウンド鍵を適用した演算を実行する演算部を有する前記（１）または（２）に記載の暗号処理装置。

[0172] （４）前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵と同一のビット数を持つ複数のラウンド鍵を生成する前記（１）～（３）いずれかに記載の暗号処理装置。

（５）前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵より少ないビット数を持つ複数のラウンド鍵構成データを生成し、前記複数のラウンド鍵構成データを複数連結して前記ラウンド演算実行部に入力するラウンド鍵と同一ビット数のラウンド鍵を生成する前記（１）～（４）いずれかに記載の暗号処理装置。

[0173] （６）前記鍵スケジュール部は、前記暗号処理部において順次実行するラウンド演算実行部に対して、ラウンド演算実行部において並列適用する複数のラウンド鍵を並列出力する記（１）～（５）いずれかに記載の暗号処理装置。

（７）前記鍵スケジュール部は、前記ラウンド演算実行部に対する鍵の選

択供給処理を行う１以上のセレクタを有する記（１）～（６）いずれかに記載の暗号処理装置。

[0174] （８）前記鍵スケジュール部は、前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、設定したグループ単位で、前記ラウンド演算実行部に対する鍵供給シーケンスの制御処理を行う記（１）～（７）いずれかに記載の暗号処理装置。

（９）前記鍵スケジュール部は、前記グループ単位のセレクタを有する記（１）～（８）いずれかに記載の暗号処理装置。

（１０）前記暗号処理部は、入力データとしての平文を暗号文に変換する暗号化処理、または、入力データとしての暗号文を平文に変換する復号処理を実行する記（１）～（９）いずれかに記載の暗号処理装置。

[0175] さらに、上記した装置およびシステムにおいて実行する処理の方法や、処理を実行させるプログラムも本開示の構成に含まれる。

[0176] また、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。例えば、プログラムは記録媒体に予め記録しておくことができる。記録媒体からコンピュータにインストールする他、ＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）、インターネットといったネットワークを介してプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。

[0177] なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるも

のには限らない。

## 産業上の利用可能性

[0178] 上述したように、本開示の一実施例の構成によれば、ラウンド鍵の供給制御により安全性の高い暗号処理装置が実現される。

具体的には、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して複数のラウンド鍵を生成する置換型鍵スケジュール部であり、暗号処理部において順次実行するラウンド演算実行部に対して、生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する。本構成により、例えば関連鍵攻撃等に対する耐性の高い安全性の高い暗号処理構成が実現される。

## 符号の説明

[0179] ７００ ＩＣモジュール

７０１ ＣＰＵ（Ｃｅｎｔｒａｌ ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ）

７０２ メモリ

７０３ 暗号処理部

７０４ 乱数生成部

７０５ 送受信部

# 請求の範囲

[請求項1] データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、

前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する暗号処理装置。

[請求項2] 前記鍵スケジュール部は、

前記ラウンド演算実行部に対する前記複数のラウンド鍵の入力シーケンスを、前記ラウンド演算実行部における複数ラウンド単位で変更する請求項１に記載の暗号処理装置。

[請求項3] 前記暗号処理部は、

前記複数のラインに分割されたデータを入力して非線形変換処理と、線形変換処理を含むＦ関数実行部と、

前記Ｆ関数実行部の出力に対して、前記ラウンド鍵を適用した演算を実行する演算部を有する請求項１に記載の暗号処理装置。

[請求項4] 前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵と同一のビット数を持つ複数のラウンド鍵を生成する請求項１に記載の暗号処理装置。

[請求項5] 前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に

入力するラウンド鍵より少ないビット数を持つ複数のラウンド鍵構成データを生成し、

前記複数のラウンド鍵構成データを複数連結して前記ラウンド演算実行部に入力するラウンド鍵と同一ビット数のラウンド鍵を生成する請求項１に記載の暗号処理装置。

［請求項6］ 前記鍵スケジュール部は、

前記暗号処理部において順次実行するラウンド演算実行部に対して、ラウンド演算実行部において並列適用する複数のラウンド鍵を並列出力する請求項１に記載の暗号処理装置。

［請求項7］ 前記鍵スケジュール部は、

前記ラウンド演算実行部に対する鍵の選択供給処理を行う１以上のセレクタを有する請求項１に記載の暗号処理装置。

［請求項8］ 前記鍵スケジュール部は、

前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、設定したグループ単位で、前記ラウンド演算実行部に対する鍵供給シーケンスの制御処理を行う請求項１に記載の暗号処理装置。

［請求項9］ 前記鍵スケジュール部は、前記グループ単位のセレクタを有する請求項８に記載の暗号処理装置。

［請求項10］ 前記暗号処理部は、

入力データとしての平文を暗号文に変換する暗号化処理、または、

入力データとしての暗号文を平文に変換する復号処理を実行する請求項１に記載の暗号処理装置。

［請求項11］ 暗号処理装置において実行する暗号処理方法であり、

暗号処理部が、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理ステップと、

鍵スケジュール部が、前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジューリングステップを実行し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力する暗号処理方法。

[請求項12] 暗号処理装置において暗号処理を実行させるプログラムであり、

暗号処理部に、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行させる暗号処理ステップと、

鍵スケジュール部に、前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジューリングステップを実行させ、

前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記鍵スケジュール部に、前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力させるプログラム。

# 補正された請求の範囲
# [2012年7月5日 (05.07.2012) 国際事務局受理]

【請求項１】（補正後）

データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、

前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力し、

前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、前記ラウンド演算実行部に対する鍵供給シーケンスを、各グループで異なる設定とする制御を行う暗号処理装置。

【請求項２】

前記鍵スケジュール部は、

前記ラウンド演算実行部に対する前記複数のラウンド鍵の入力シーケンスを、前記ラウンド演算実行部における複数ラウンド単位で変更する請求項１に記載の暗号処理装置。

【請求項３】

前記暗号処理部は、

前記複数のラインに分割されたデータを入力して非線形変換処理と、線形変換処理を含むＦ関数実行部と、

前記Ｆ関数実行部の出力に対して、前記ラウンド鍵を適用した演算を実行する演算部を有する請求項１に記載の暗号処理装置。

【請求項４】

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵と同一のビット数を持つ複数のラウンド鍵を生成する請求項１に記載の暗号処理装置。

【請求項５】

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して前記ラウンド演算実行部に入力するラウンド鍵より少ないビット数を持つ複数のラウンド鍵構成データを生成し、

前記複数のラウンド鍵構成データを複数連結して前記ラウンド演算実行部に入力するラウンド鍵と同一ビット数のラウンド鍵を生成する請求項１に記載の暗号処理装置。

【請求項６】（補正後）

前記鍵スケジュール部は、

前記暗号処理部において順次実行するラウンド演算実行部に対して、ラウンド演算実行部において適用する複数のラウンド鍵を並列出力する請求項１に記載の暗号処理装置。

【請求項７】

前記鍵スケジュール部は、

前記ラウンド演算実行部に対する鍵の選択供給処理を行なう１以上のセレクタを有する請求項１に記載の暗号処理装置。

【請求項８】（削除）

【請求項９】（補正後）

前記鍵スケジュール部は、前記グループ各々に対応する個別のセレクタを有する請求項１に記載の暗号処理装置。

【請求項１０】

前記暗号処理部は、

入力データとしての平文を暗号文に変換する暗号化処理、または、

入力データとしての暗号文を平文に変換する復号処理を実行する請求項１に記載の暗号処理装置。

【請求項１１】（補正後）

暗号処理装置において実行する暗号処理方法であり、

暗号処理部が、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理ステップと、

鍵スケジュール部が、前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジューリングステップを実行し、

前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力し、

前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、前記ラウンド演算実行部に対する鍵供給シーケンスを、各グループで異なる設定とする制御を行う暗号処理方法。

【請求項１２】（補正後）

暗号処理装置において暗号処理を実行させるプログラムであり、

暗号処理部に、データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行させる暗号処理ステップと、

鍵スケジュール部に、前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジューリングステップを実行させ、

前記鍵スケジュール部は、予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

前記鍵スケジュール部に、前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力させ、

前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、前記ラウンド演算実行部に対する鍵供給シーケンスを、各グループで異なる設定と

する制御を行わせるプログラム。

【請求項１３】（追加）

前記暗号処置装置は、前記暗号処理部と前記鍵スケジュール部から構成される一般化Ｆｅｉｓｔｅｌ構造をもつ、請求項１に記載の暗号処理装置。

【請求項１４】（追加）

前記暗号処置装置は、前記暗号処理部と前記鍵スケジュール部から構成される４系列一般化Ｆｅｉｓｔｅｌ構造をもつ、請求項１に記載の暗号処理装置。

【請求項１５】（追加）

前記暗号処置装置は、前記暗号処理部と前記鍵スケジュール部から構成されるＴｙｐｅ－２一般化Ｆｅｉｓｔｅｌ構造をもつ、請求項１に記載の暗号処理装置。

【請求項１６】（追加）

プログラムを実行するプロセッサと、

前記プログラムを格納するメモリと、

データ処理対象となるデータの構成ビットを複数のラインに分割して入力し、各ラインのデータに対してラウンド関数を適用したデータ変換処理をラウンド演算として繰り返して実行する暗号処理部と、

前記暗号処理部のラウンド演算実行部に対して、ラウンド鍵を出力する鍵スケジュール部を有し、

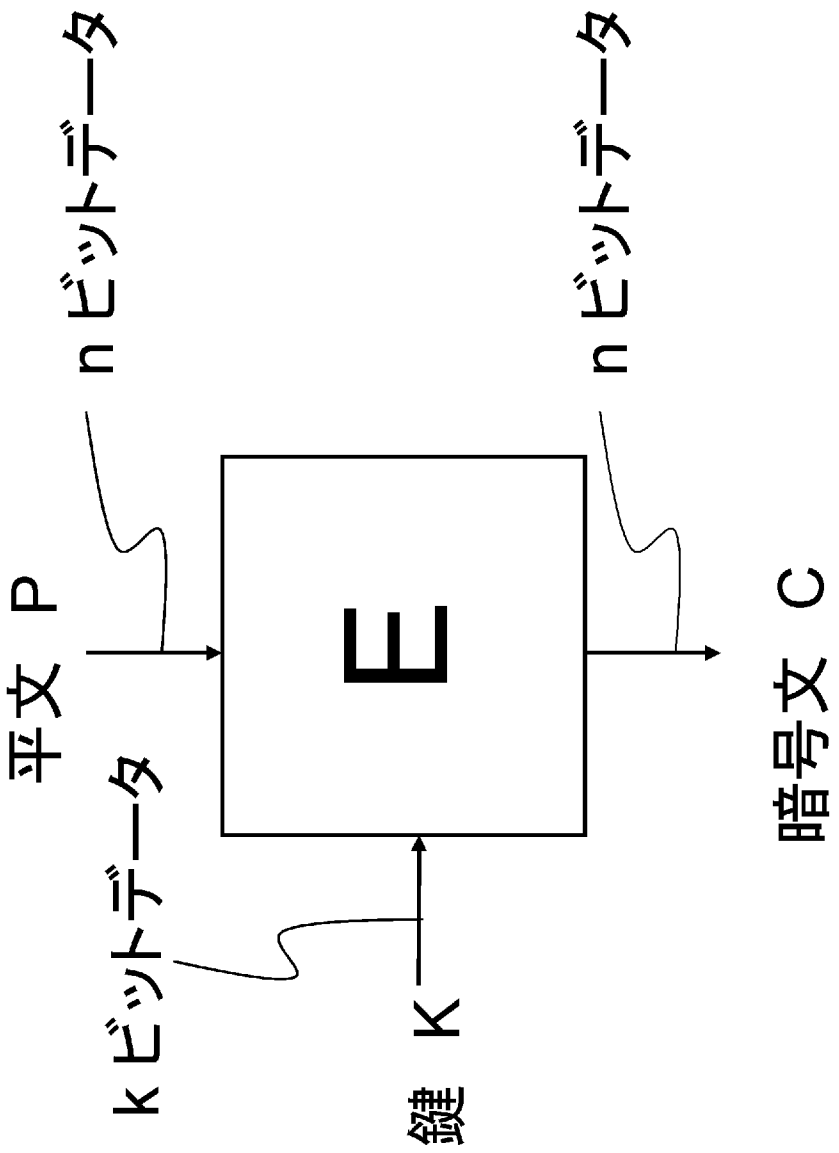
前記鍵スケジュール部は、

予め保持する秘密鍵を複数個に分割して複数のラウンド鍵またはラウンド鍵構成データを生成する置換型鍵スケジュール部であり、

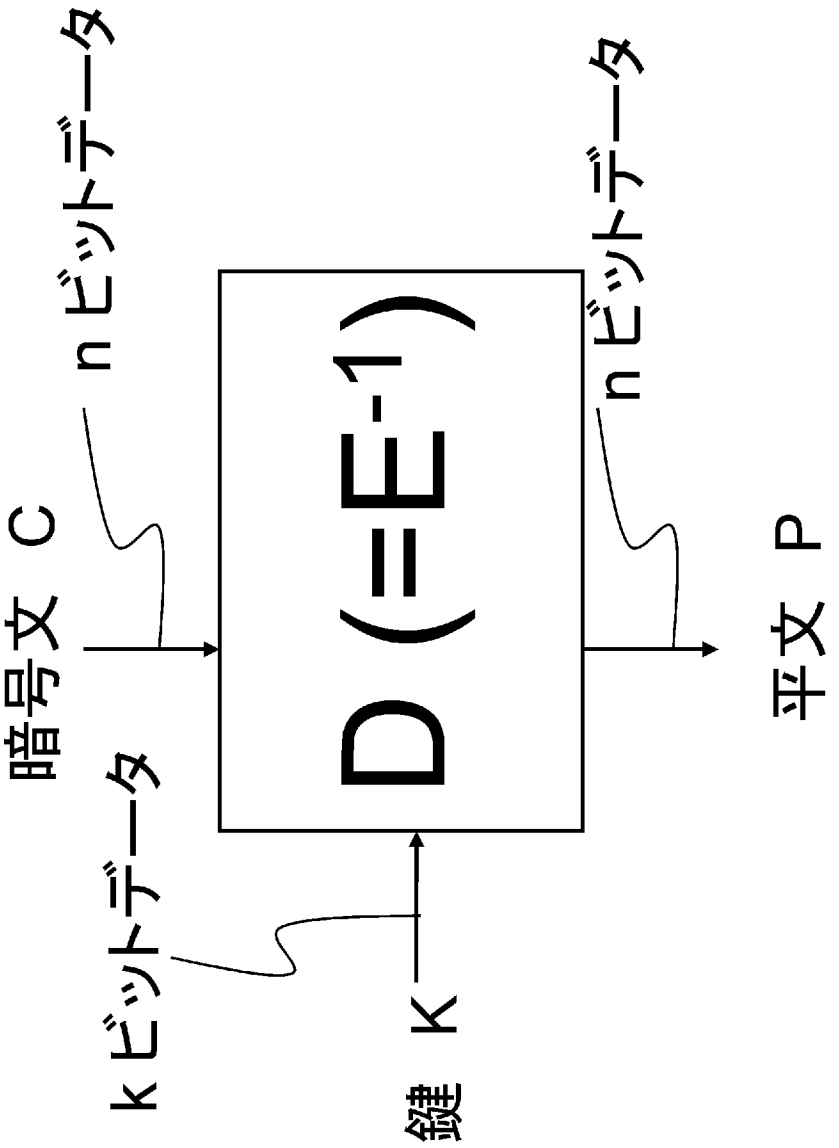
前記暗号処理部において順次実行するラウンド演算実行部に対して、前記複数のラウンド鍵、または前記ラウンド鍵構成データを結合して生成した複数のラウンド鍵を一定のシーケンスの繰り返しとならない設定で出力するし、

前記複数のラウンド鍵または前記ラウンド鍵構成データを区分した複数のグループを設定し、前記ラウンド演算実行部に対する鍵供給シーケンスを、各グループで異なる設定とする制御を行う情報処理装置。

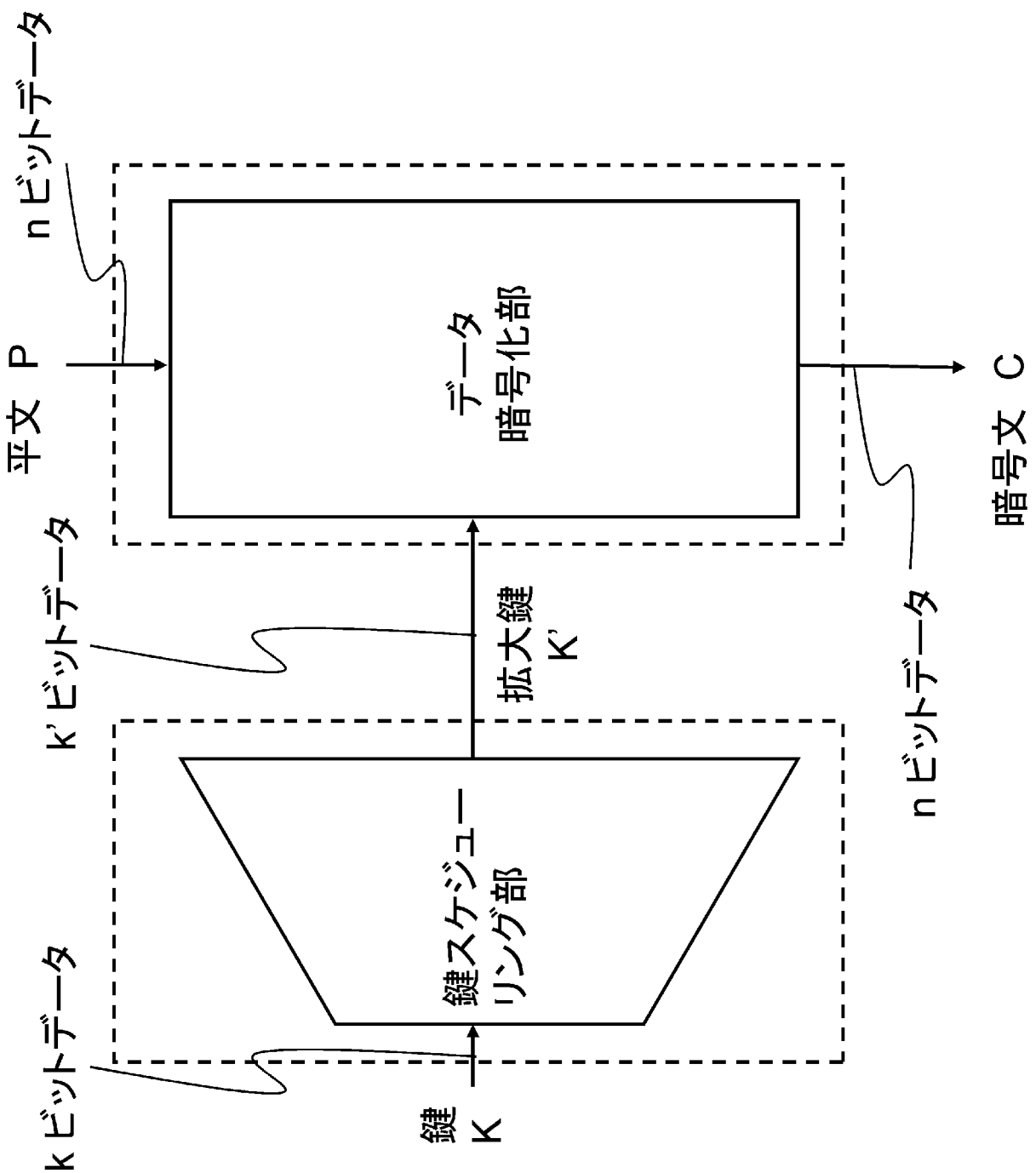
[図1]



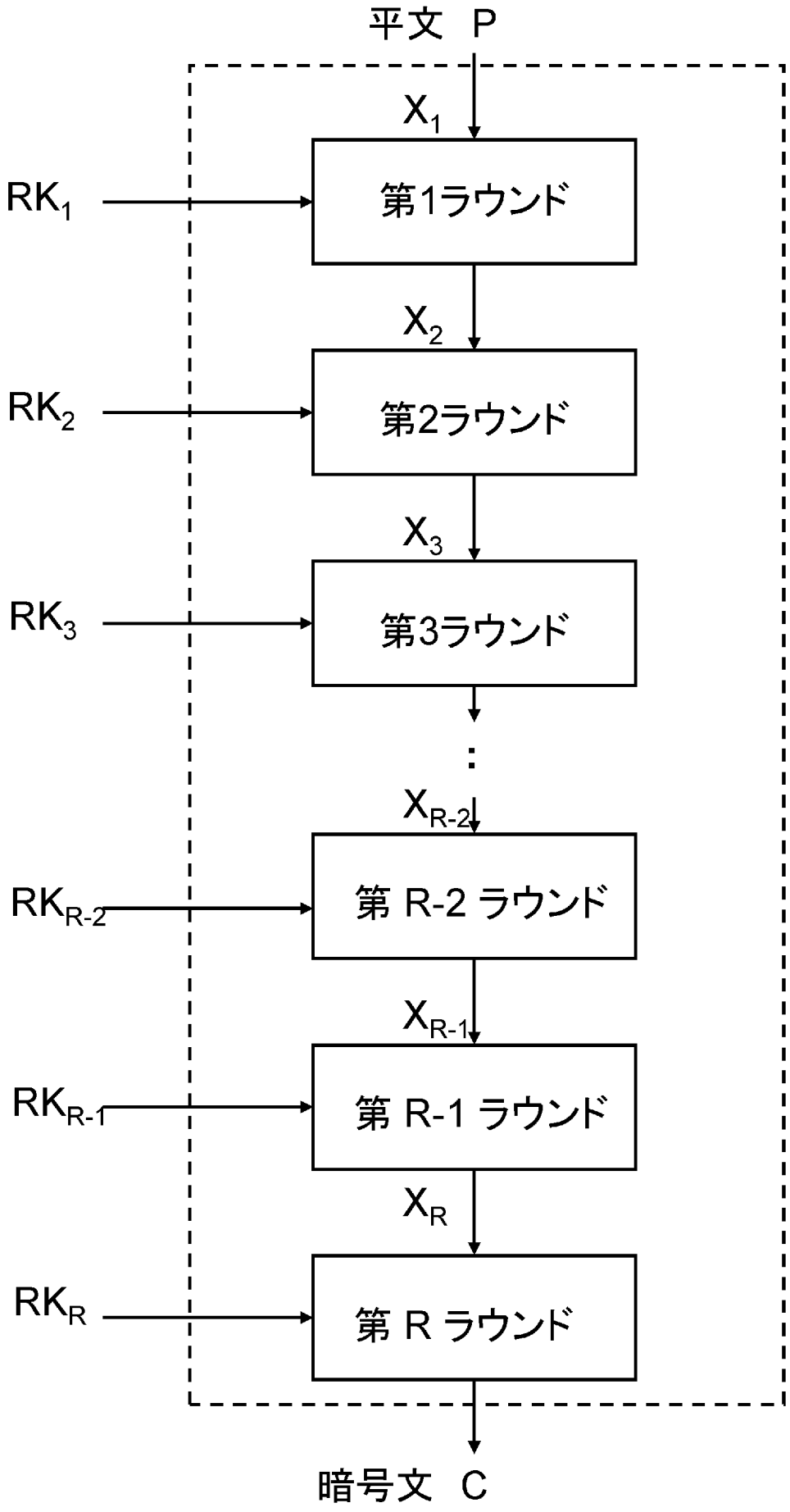
[図2]



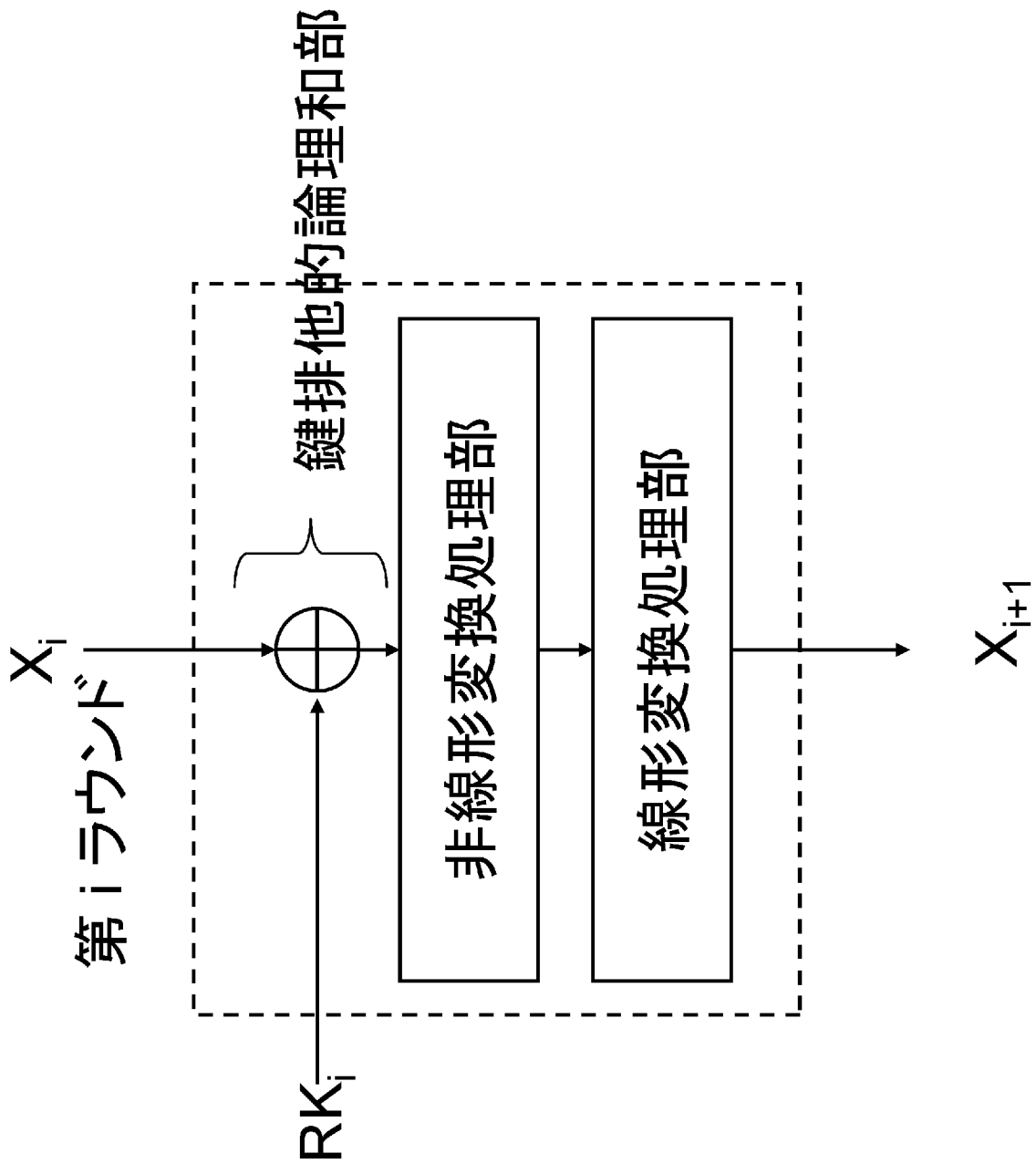
[図3]


nビットデータ
平文 P
データ
暗号化部
暗号文 C
K'ビットデータ
拡大鍵
K'
nビットデータ
鍵スケジュー
リング部
kビットデータ
鍵
K

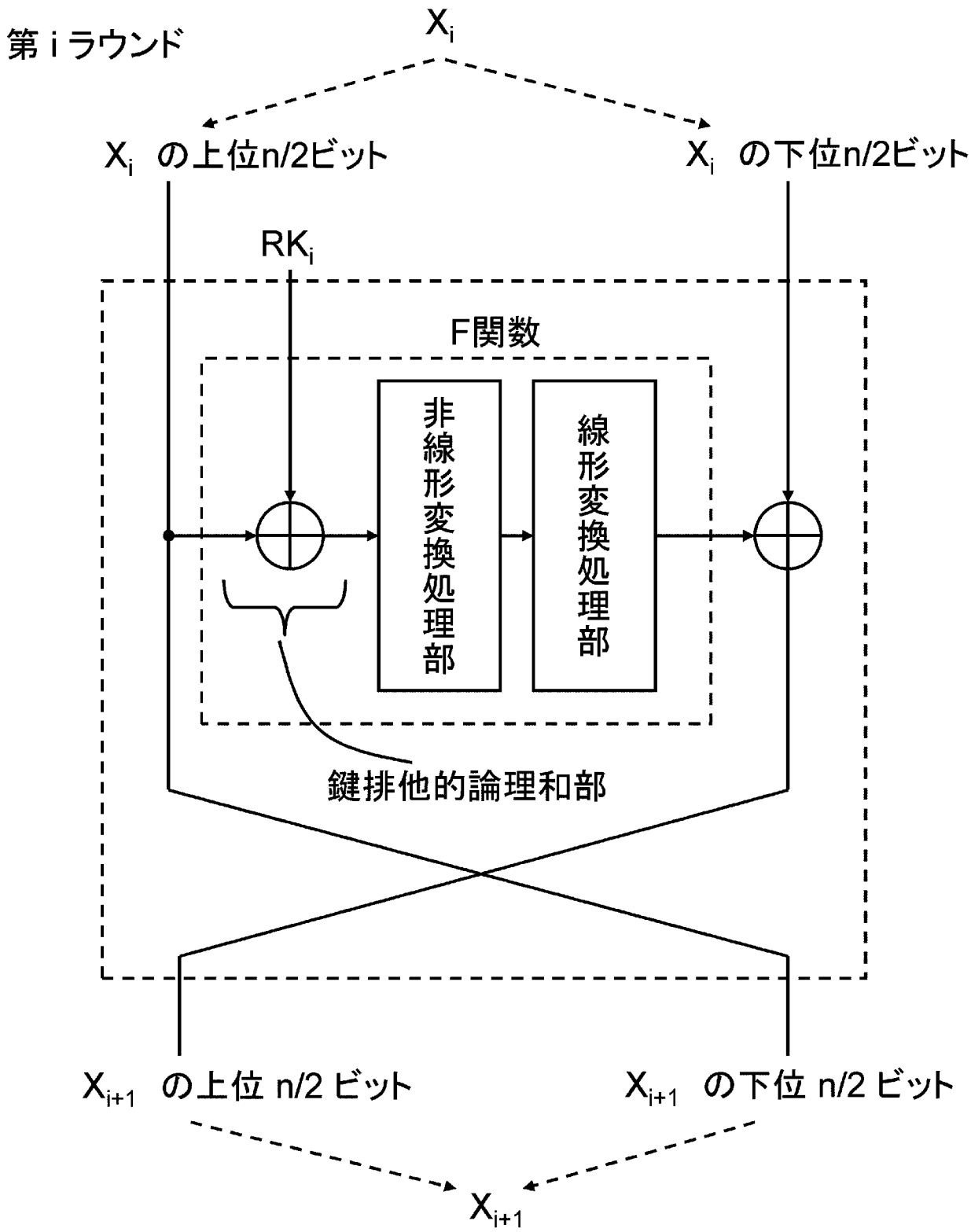
[図4]


平文 P
X1
RK1
第1ラウンド
X2
RK2
第2ラウンド
X3
RK3
第3ラウンド
XR-2
RKR-2
第 R-2 ラウンド
XR-1
RKR-1
第 R-1 ラウンド
XR
RKR
第 R ラウンド
暗号文 C

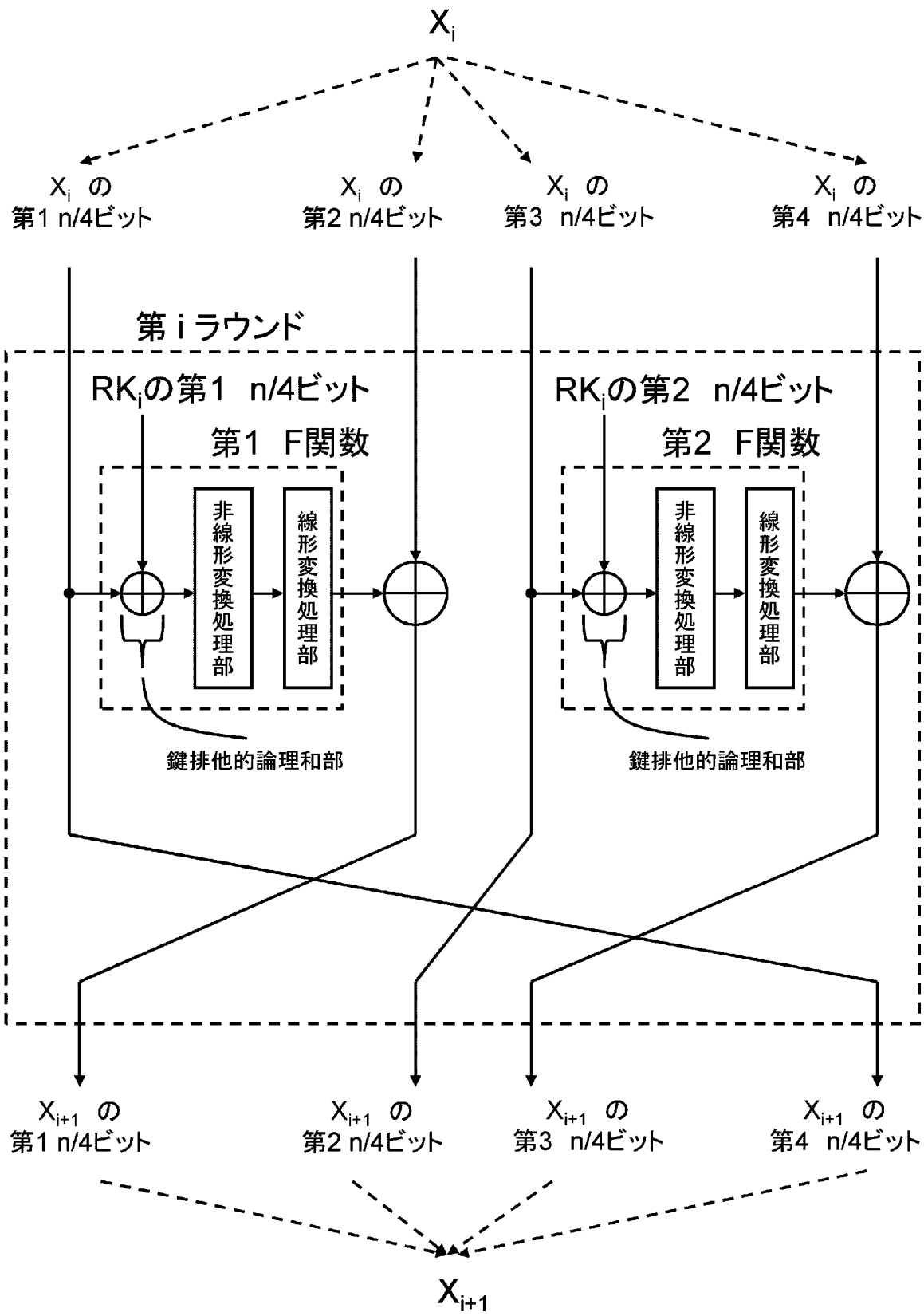
[図5]



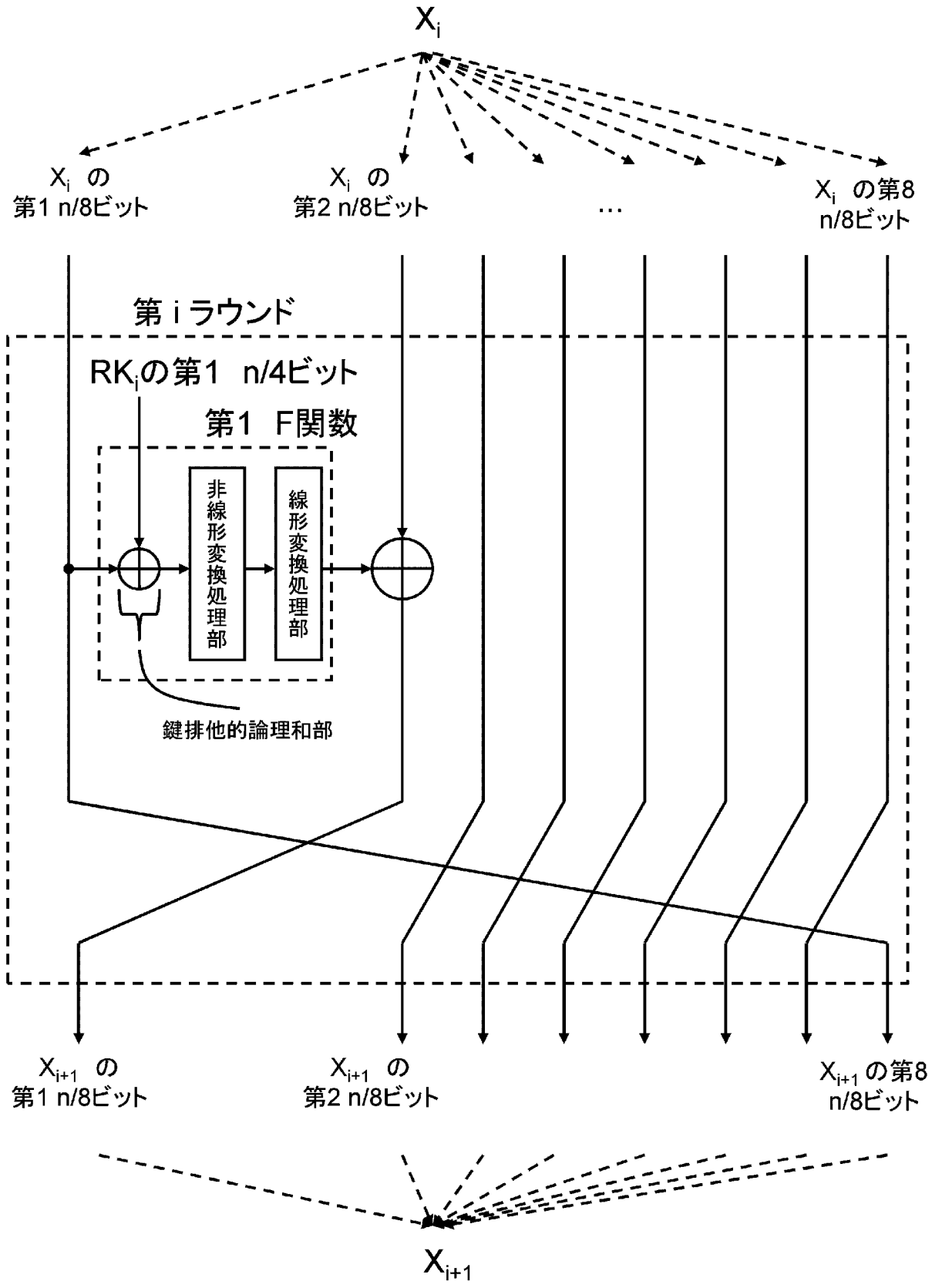
[図6]


第 i ラウンド
$X_i$
$X_i$ の上位n/2ビット
$X_i$ の下位n/2ビット
$RK_i$
F関数
非線形変換処理部
線形変換処理部
鍵排他的論理和部
$X_{i+1}$ の上位 n/2 ビット
$X_{i+1}$ の下位 n/2 ビット
$X_{i+1}$

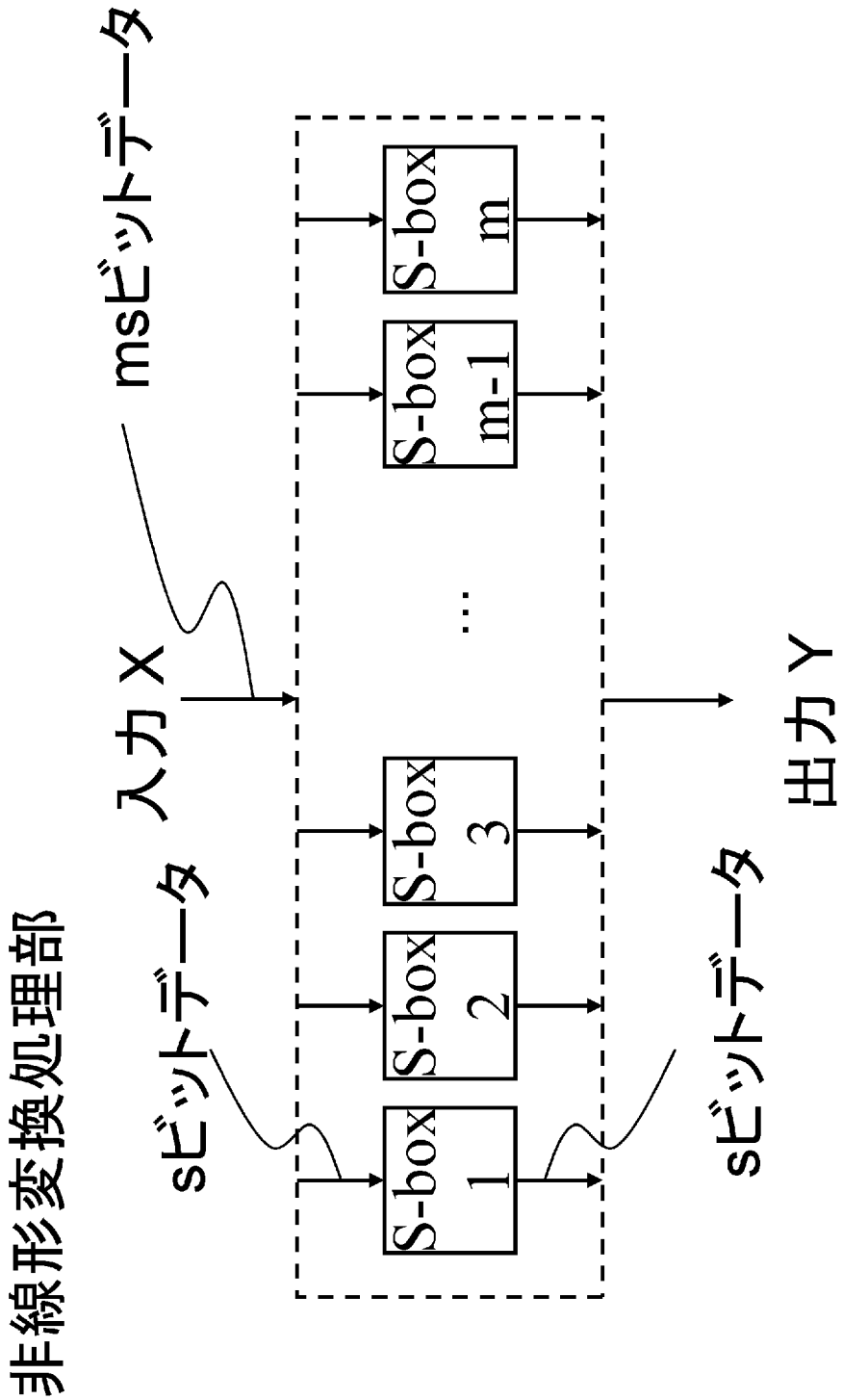
[図7]


$X_i$
$X_i$ の 第1 n/4ビット
$X_i$ の 第2 n/4ビット
$X_i$ の 第3 n/4ビット
$X_i$ の 第4 n/4ビット
第 i ラウンド
$RK_i$の第1 n/4ビット
第1 F関数
非線形変換処理部
線形変換処理部
鍵排他的論理和部
$RK_i$の第2 n/4ビット
第2 F関数
非線形変換処理部
線形変換処理部
鍵排他的論理和部
$X_{i+1}$ の 第1 n/4ビット
$X_{i+1}$ の 第2 n/4ビット
$X_{i+1}$ の 第3 n/4ビット
$X_{i+1}$ の 第4 n/4ビット
$X_{i+1}$

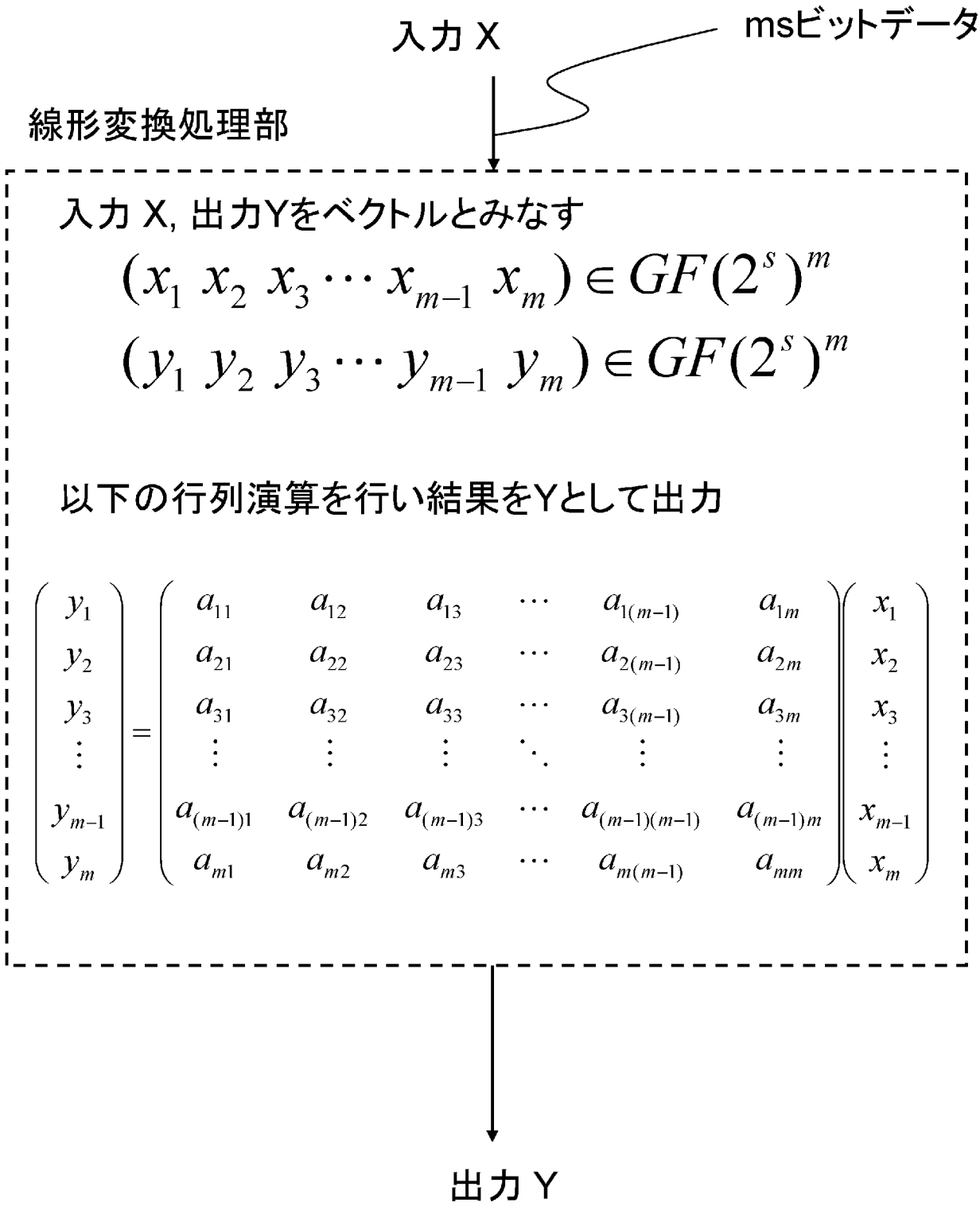
[図8]


$X_i$
$X_i$ の
第1 n/8ビット
$X_i$ の
第2 n/8ビット
…
$X_i$ の第8
n/8ビット
第 i ラウンド
$RK_i$の第1 n/4ビット
第1 F関数
非線形変換処理部
線形変換処理部
鍵排他的論理和部
$X_{i+1}$ の
第1 n/8ビット
$X_{i+1}$ の
第2 n/8ビット
$X_{i+1}$ の第8
n/8ビット
$X_{i+1}$

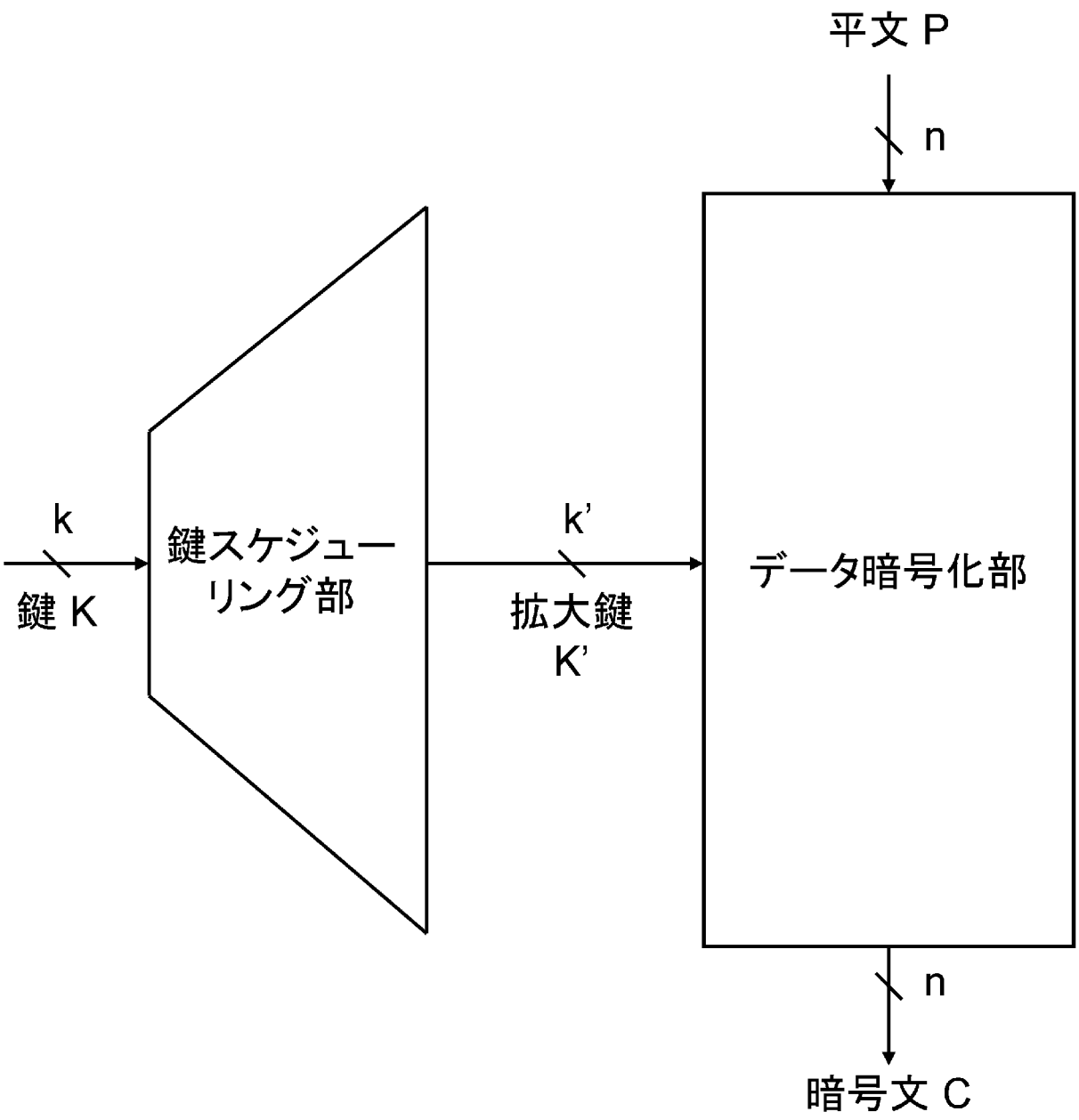
[図9]


非線形変換処理部
入力 X
msビットデータ
sビットデータ
S-box 1
S-box 2
S-box 3
S-box m-1
S-box m
出力 Y
sビットデータ

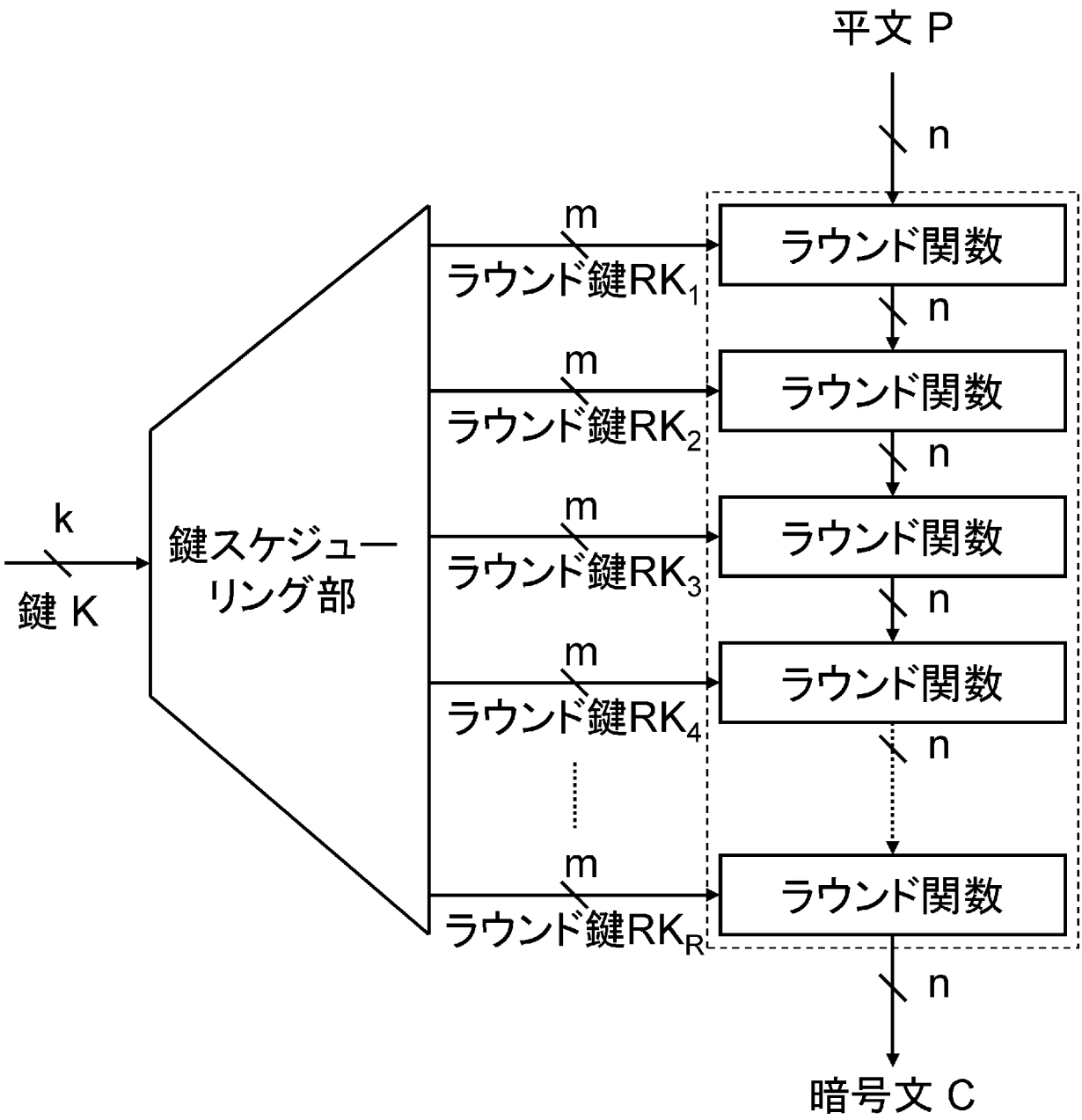
[図10]


入力 X
msビットデータ
線形変換処理部
入力 X, 出力Yをベクトルとみなす
$(x_1\ x_2\ x_3 \cdots x_{m-1}\ x_m) \in GF(2^s)^m$
$(y_1\ y_2\ y_3 \cdots y_{m-1}\ y_m) \in GF(2^s)^m$
以下の行列演算を行い結果をYとして出力
$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{m-1} \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1(m-1)} & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2(m-1)} & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3(m-1)} & a_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{(m-1)1} & a_{(m-1)2} & a_{(m-1)3} & \cdots & a_{(m-1)(m-1)} & a_{(m-1)m} \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{m(m-1)} & a_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix}$
出力 Y

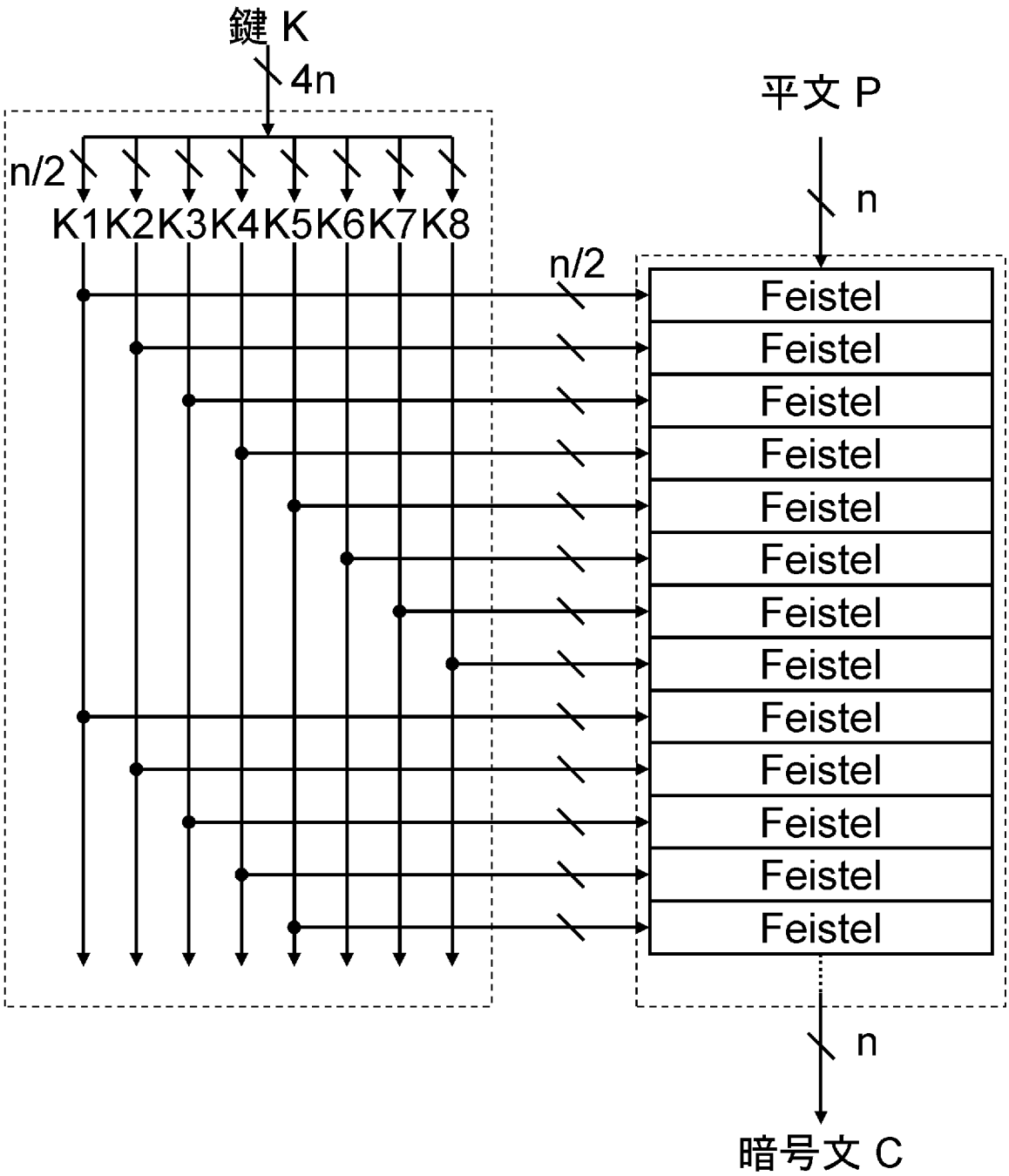
[図11]



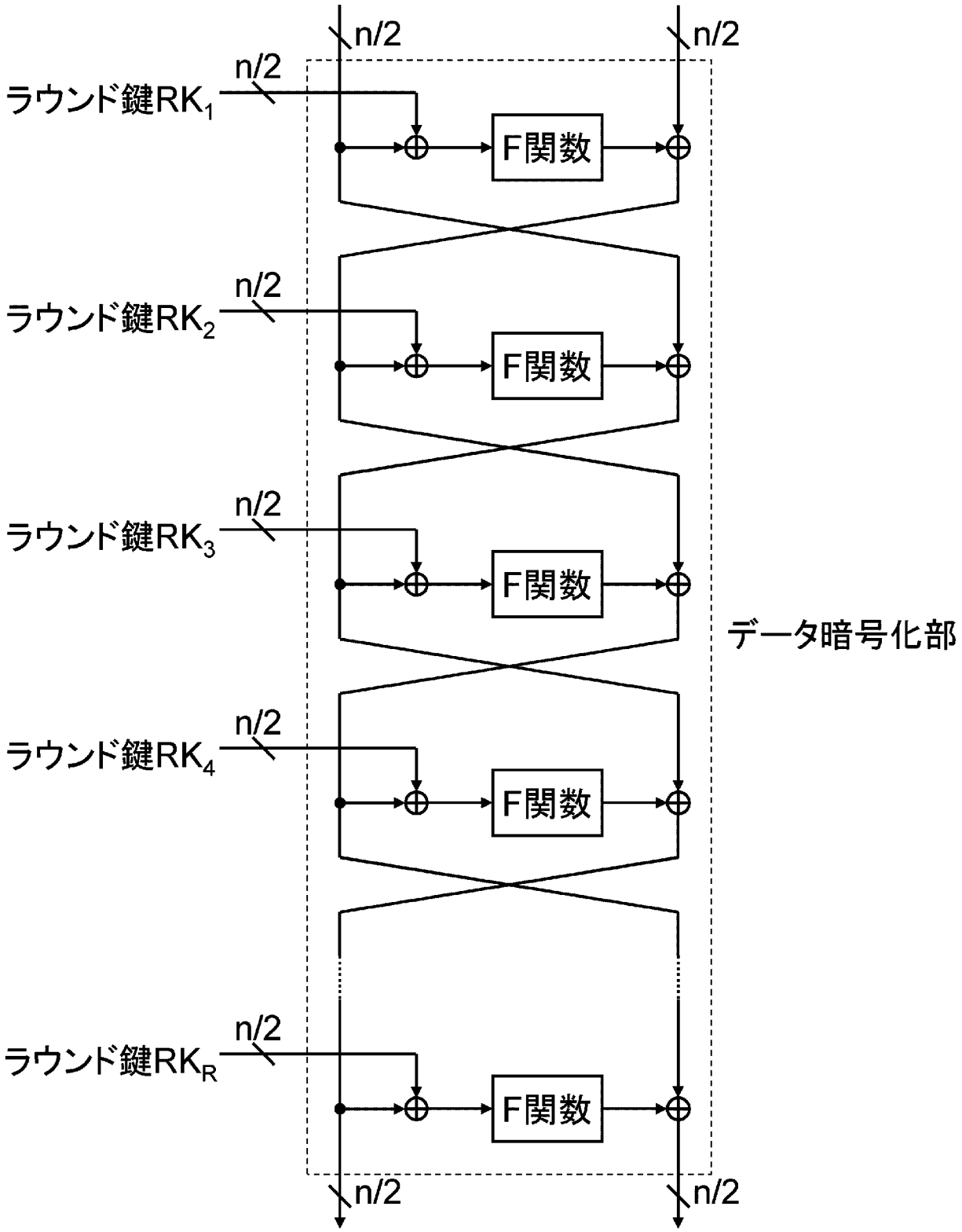
[図12]


平文 P
n
m
ラウンド鍵RK1
ラウンド関数
n
m
ラウンド鍵RK2
ラウンド関数
n
k
鍵 K
鍵スケジューリング部
m
ラウンド鍵RK3
ラウンド関数
n
m
ラウンド鍵RK4
ラウンド関数
n
m
ラウンド鍵RKR
ラウンド関数
n
暗号文 C

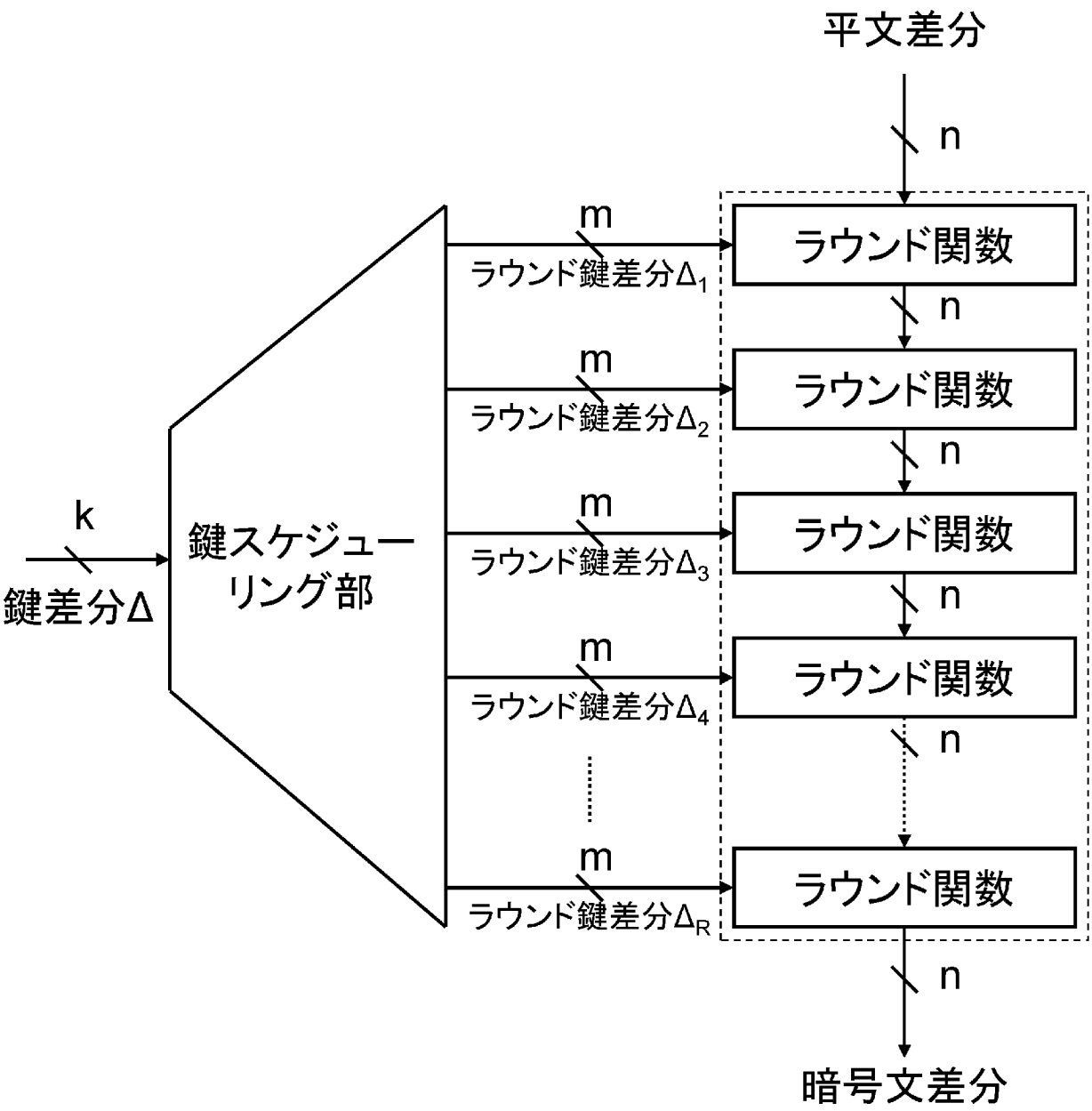
[図13]


鍵 K
4n
n/2
K1K2K3K4K5K6K7K8
n/2
平文 P
n
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
Feistel
n
暗号文 C

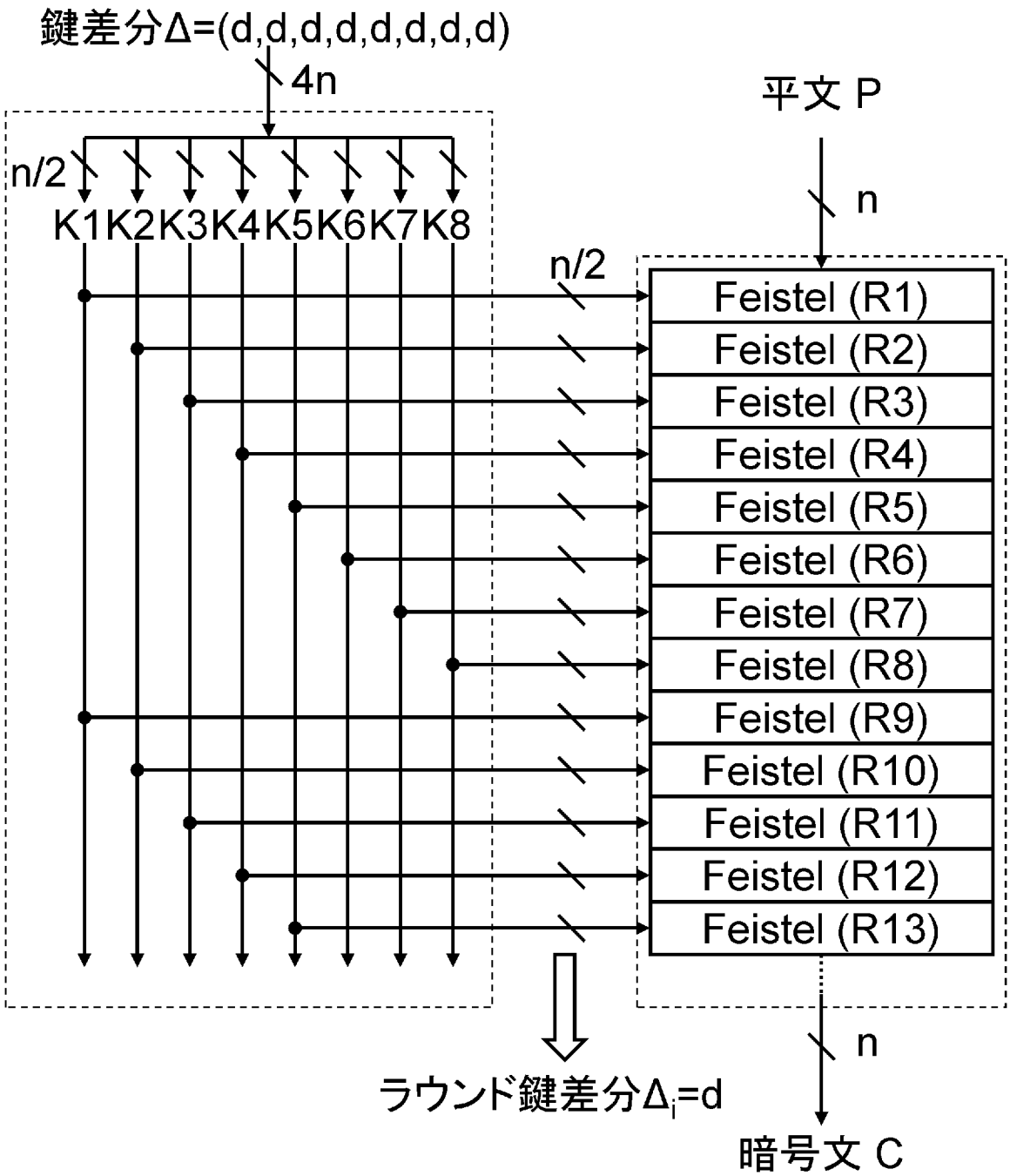
[図14]


n/2
n/2
n/2
ラウンド鍵RK1
F関数
n/2
ラウンド鍵RK2
F関数
n/2
ラウンド鍵RK3
F関数
データ暗号化部
n/2
ラウンド鍵RK4
F関数
n/2
ラウンド鍵RKR
F関数
n/2
n/2

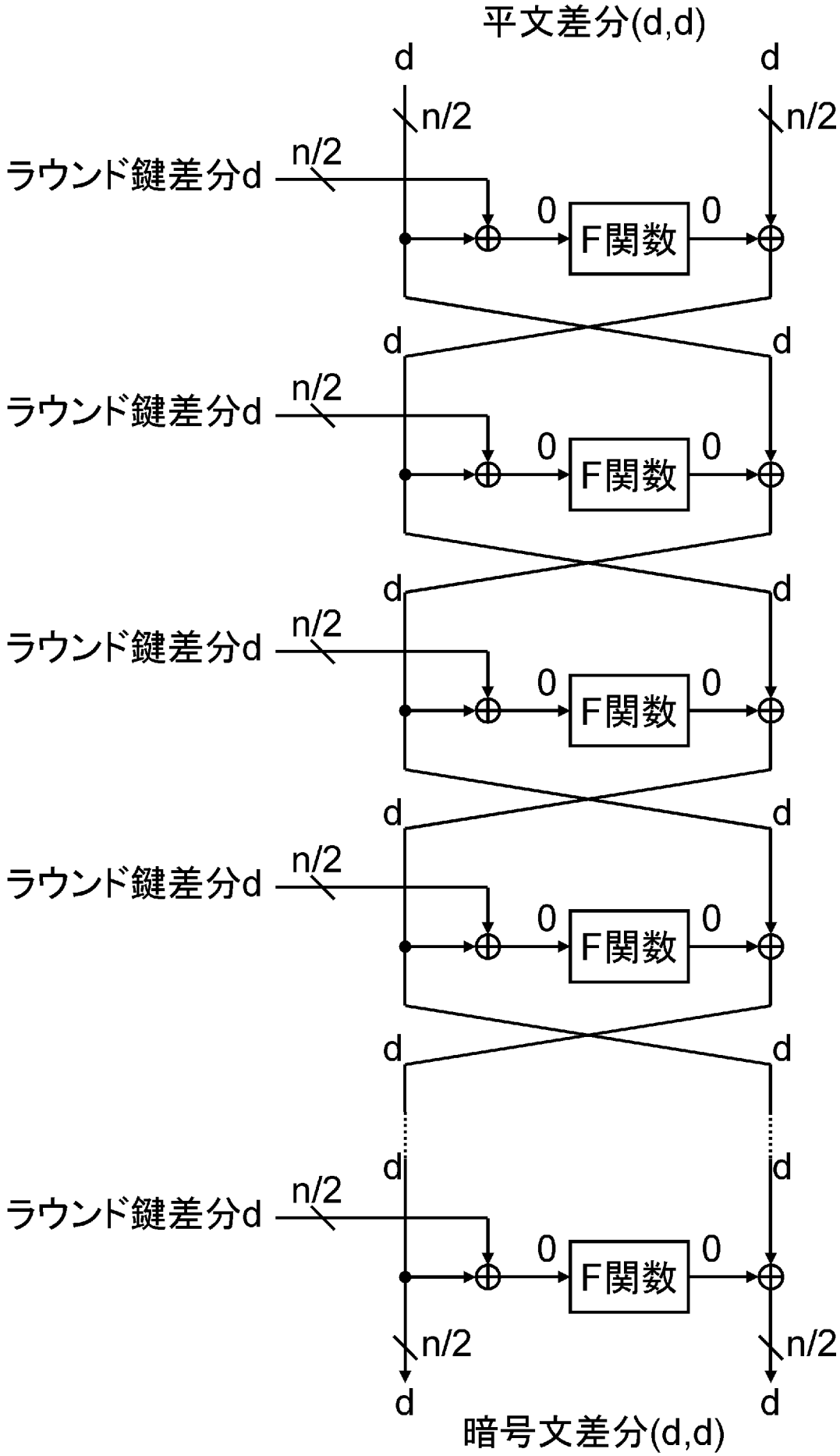
[図15]


平文差分
n
m
ラウンド鍵差分Δ1
ラウンド関数
n
m
ラウンド鍵差分Δ2
ラウンド関数
n
k
鍵差分Δ
鍵スケジューリング部
m
ラウンド鍵差分Δ3
ラウンド関数
n
m
ラウンド鍵差分Δ4
ラウンド関数
n
m
ラウンド鍵差分ΔR
ラウンド関数
n
暗号文差分

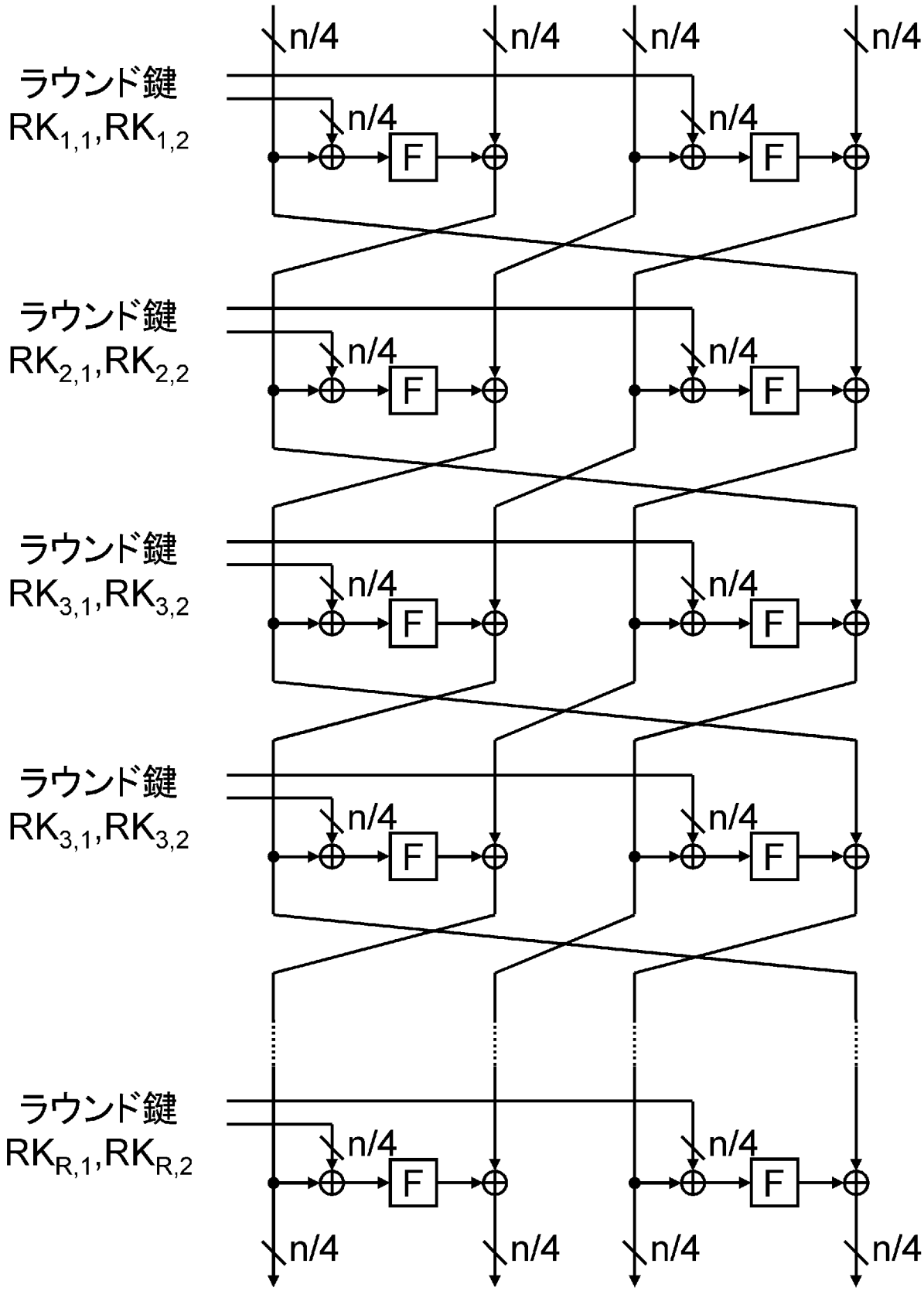
[図16]


鍵差分Δ=(d,d,d,d,d,d,d,d)
4n
n/2
K1K2K3K4K5K6K7K8
n/2
平文 P
n
Feistel (R1)
Feistel (R2)
Feistel (R3)
Feistel (R4)
Feistel (R5)
Feistel (R6)
Feistel (R7)
Feistel (R8)
Feistel (R9)
Feistel (R10)
Feistel (R11)
Feistel (R12)
Feistel (R13)
n
ラウンド鍵差分Δi=d
暗号文 C

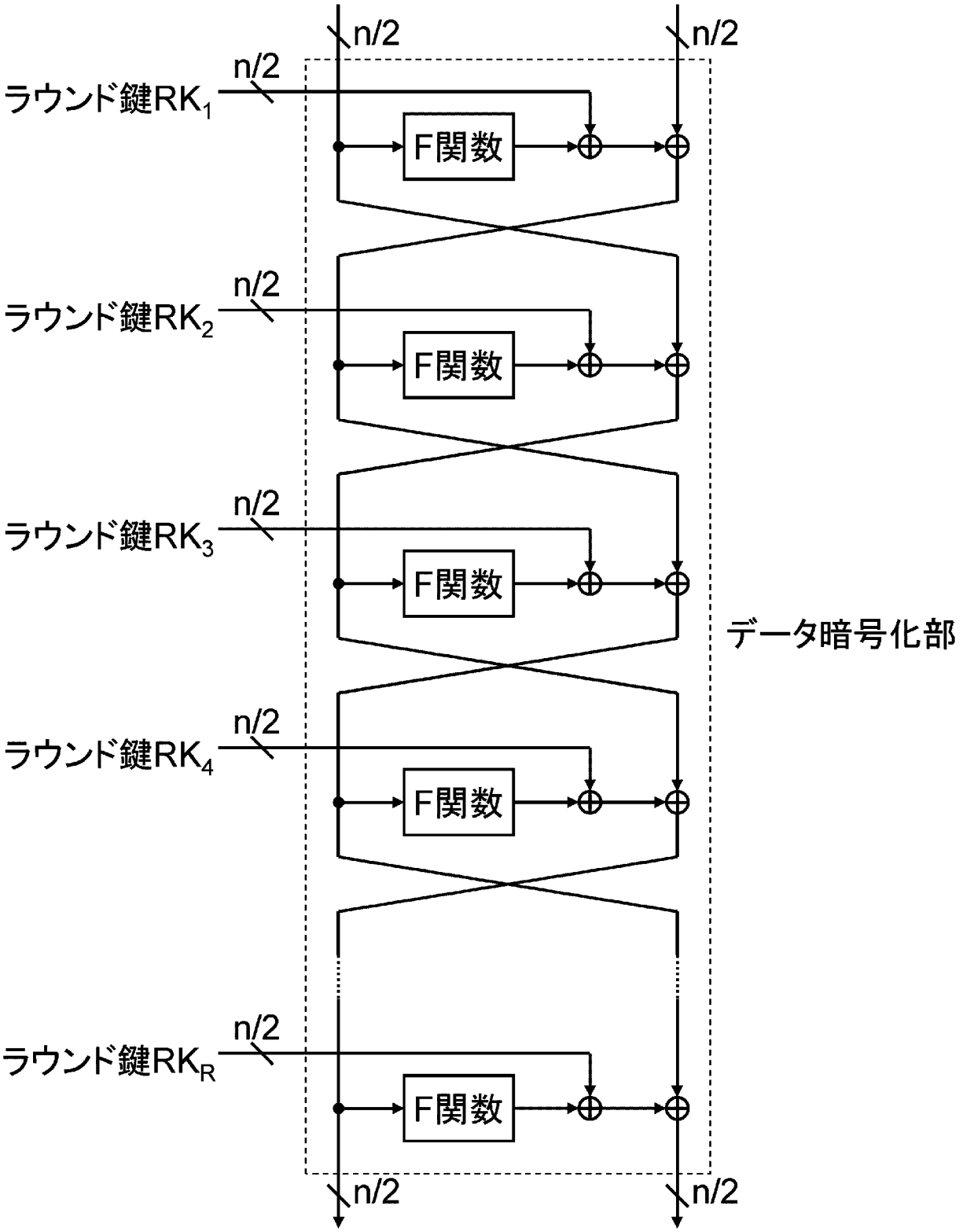
[図17]



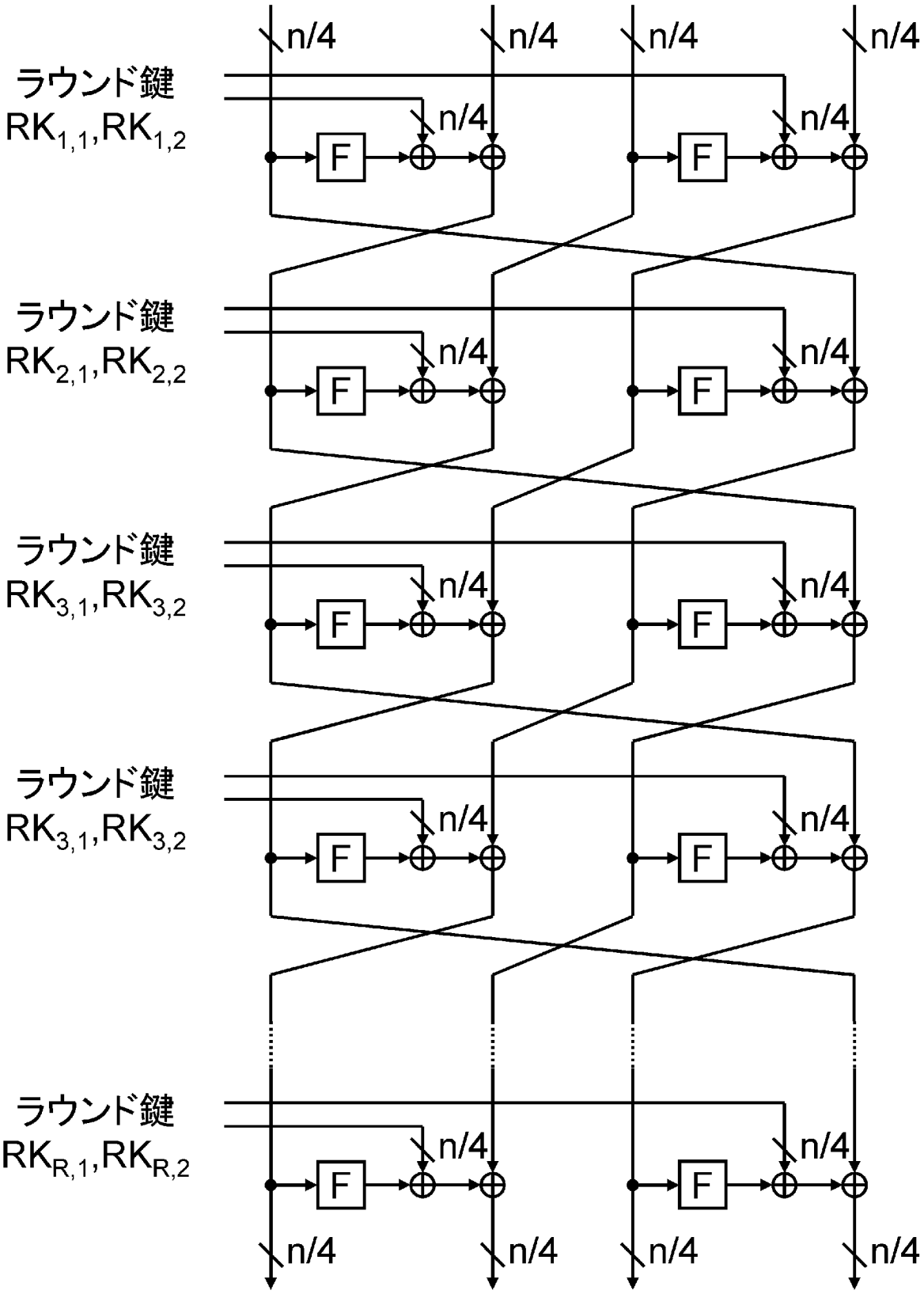
[図18]


n/4
n/4
n/4
n/4
ラウンド鍵
RK1,1,RK1,2
n/4
F
n/4
F
ラウンド鍵
RK2,1,RK2,2
n/4
F
n/4
F
ラウンド鍵
RK3,1,RK3,2
n/4
F
n/4
F
ラウンド鍵
RK3,1,RK3,2
n/4
F
n/4
F
ラウンド鍵
RKR,1,RKR,2
n/4
F
n/4
F
n/4
n/4
n/4
n/4

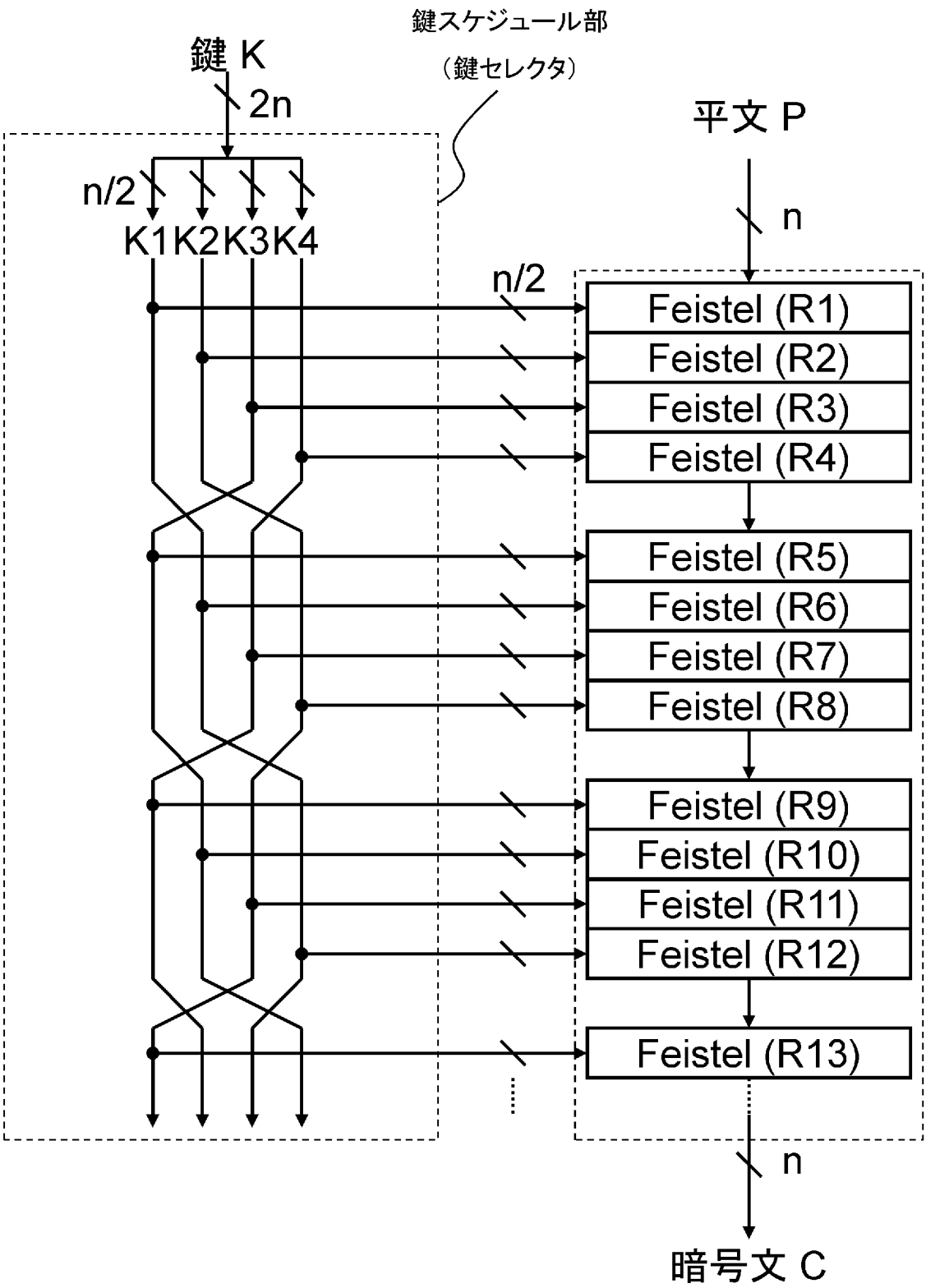
[図19]


n/2
n/2
n/2
ラウンド鍵RK1
F関数
n/2
ラウンド鍵RK2
F関数
n/2
ラウンド鍵RK3
F関数
データ暗号化部
n/2
ラウンド鍵RK4
F関数
n/2
ラウンド鍵RKR
F関数
n/2
n/2

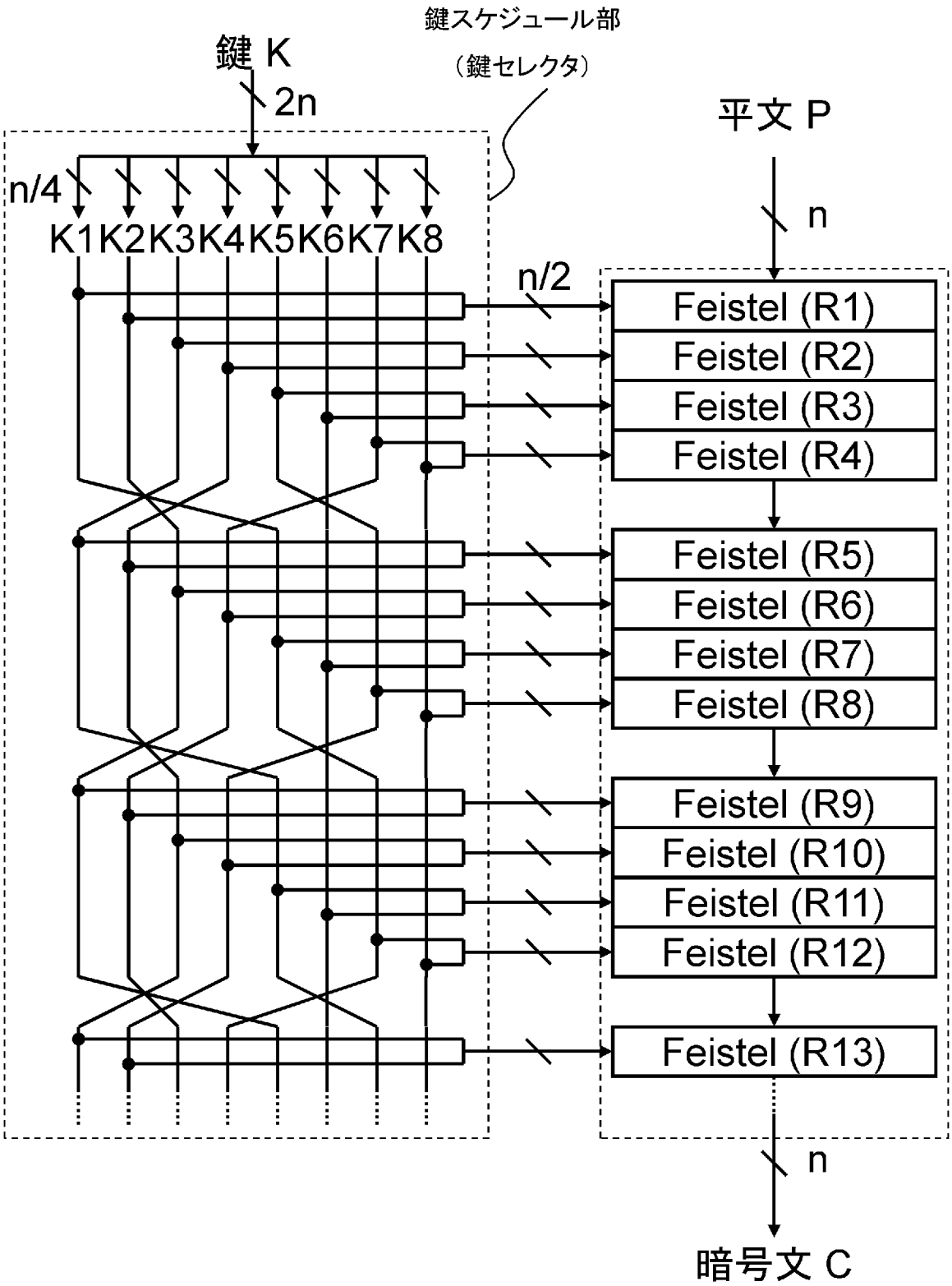
[図20]


n/4
n/4
n/4
n/4
ラウンド鍵
RK1,1,RK1,2
F
n/4
n/4
F
ラウンド鍵
RK2,1,RK2,2
ラウンド鍵
RK3,1,RK3,2
ラウンド鍵
RK3,1,RK3,2
ラウンド鍵
RKR,1,RKR,2
n/4
n/4
n/4
n/4

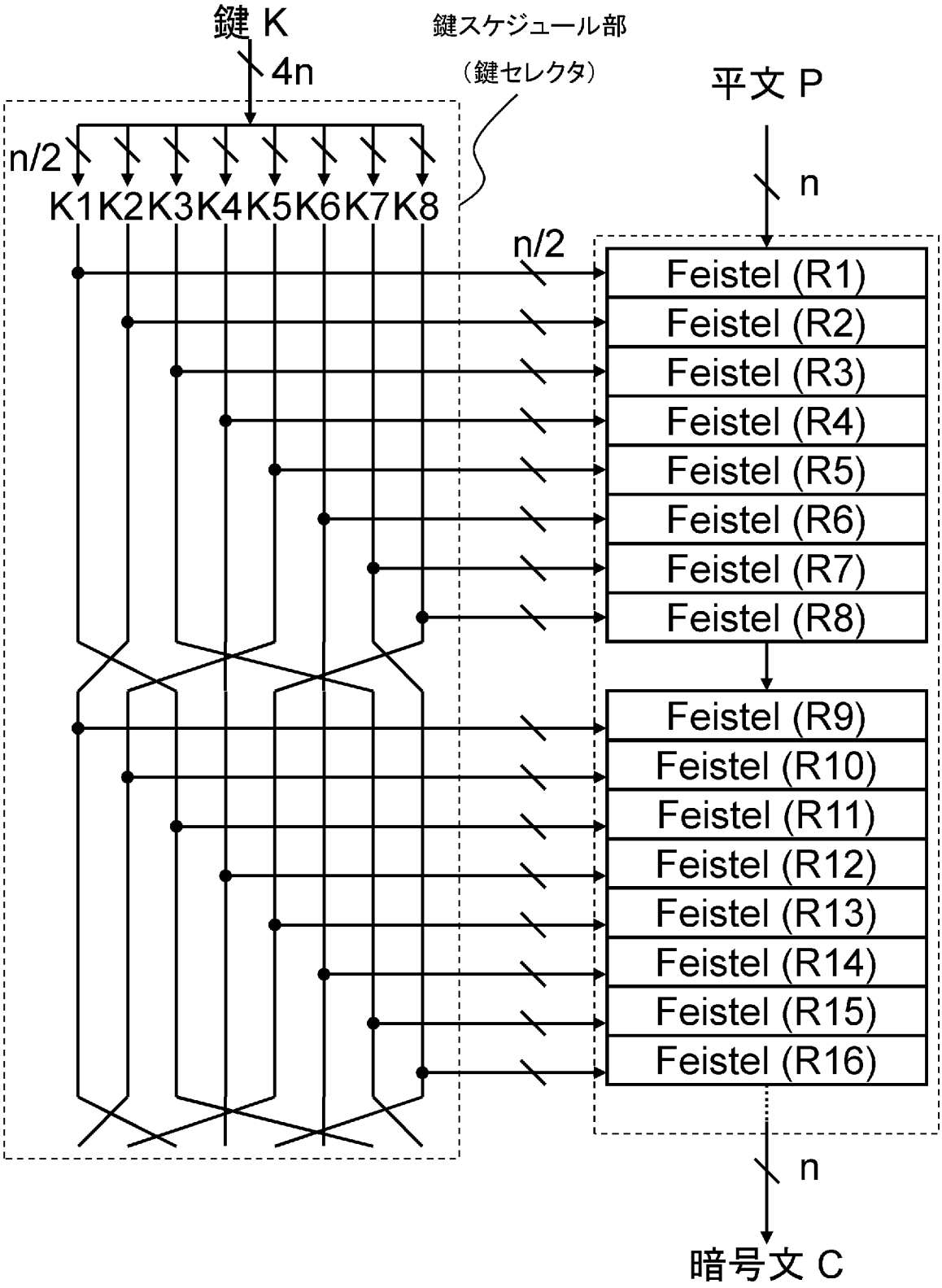
[図21]


鍵スケジュール部
（鍵セレクタ）
鍵 K
2n
n/2
K1K2K3K4
平文 P
n
n/2
Feistel (R1)
Feistel (R2)
Feistel (R3)
Feistel (R4)
Feistel (R5)
Feistel (R6)
Feistel (R7)
Feistel (R8)
Feistel (R9)
Feistel (R10)
Feistel (R11)
Feistel (R12)
Feistel (R13)
n
暗号文 C

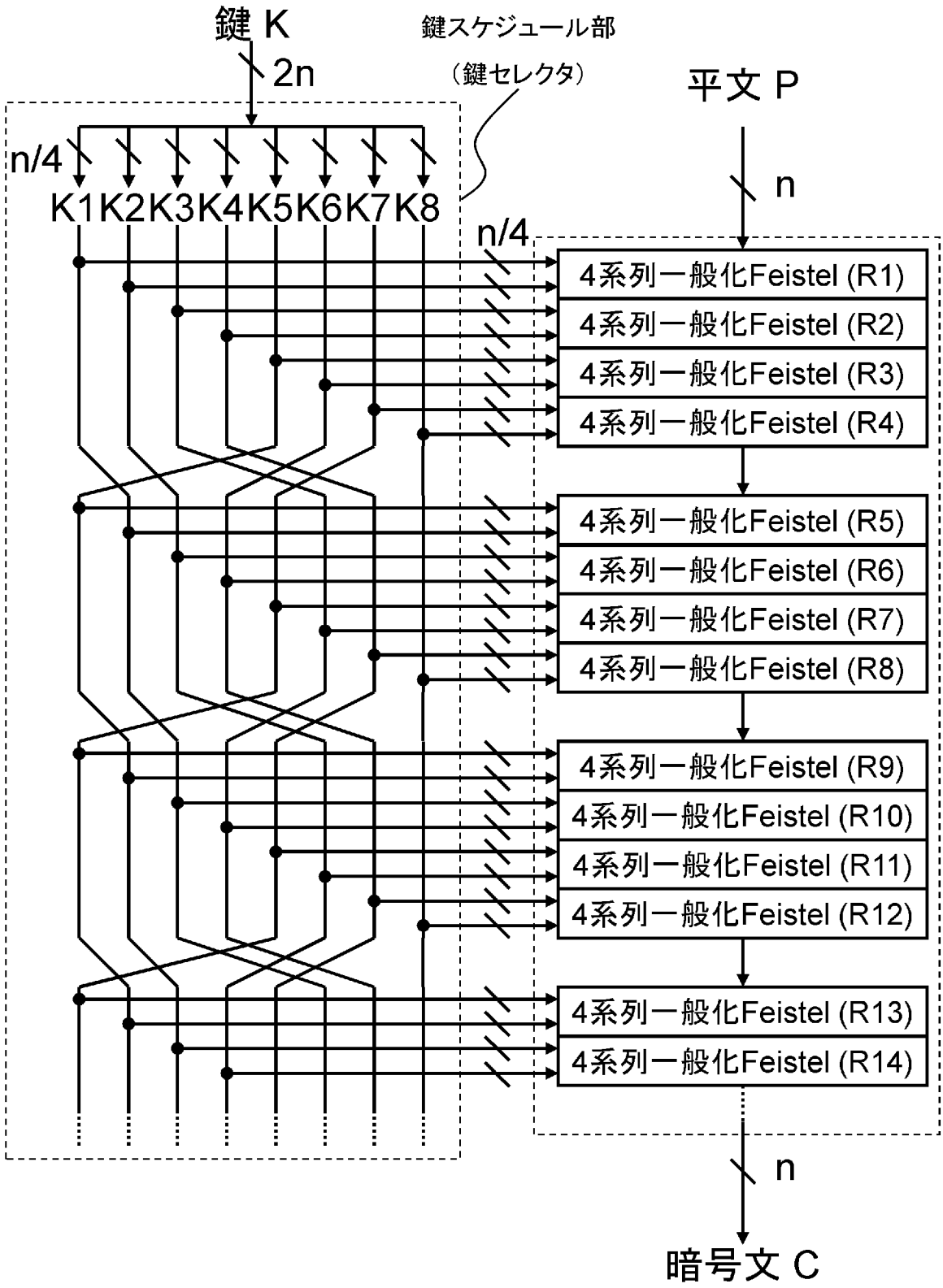
[図22]


鍵スケジュール部
（鍵セレクタ）
鍵 K
2n
n/4
K1K2K3K4K5K6K7K8
n/2
平文 P
n
Feistel (R1)
Feistel (R2)
Feistel (R3)
Feistel (R4)
Feistel (R5)
Feistel (R6)
Feistel (R7)
Feistel (R8)
Feistel (R9)
Feistel (R10)
Feistel (R11)
Feistel (R12)
Feistel (R13)
n
暗号文 C

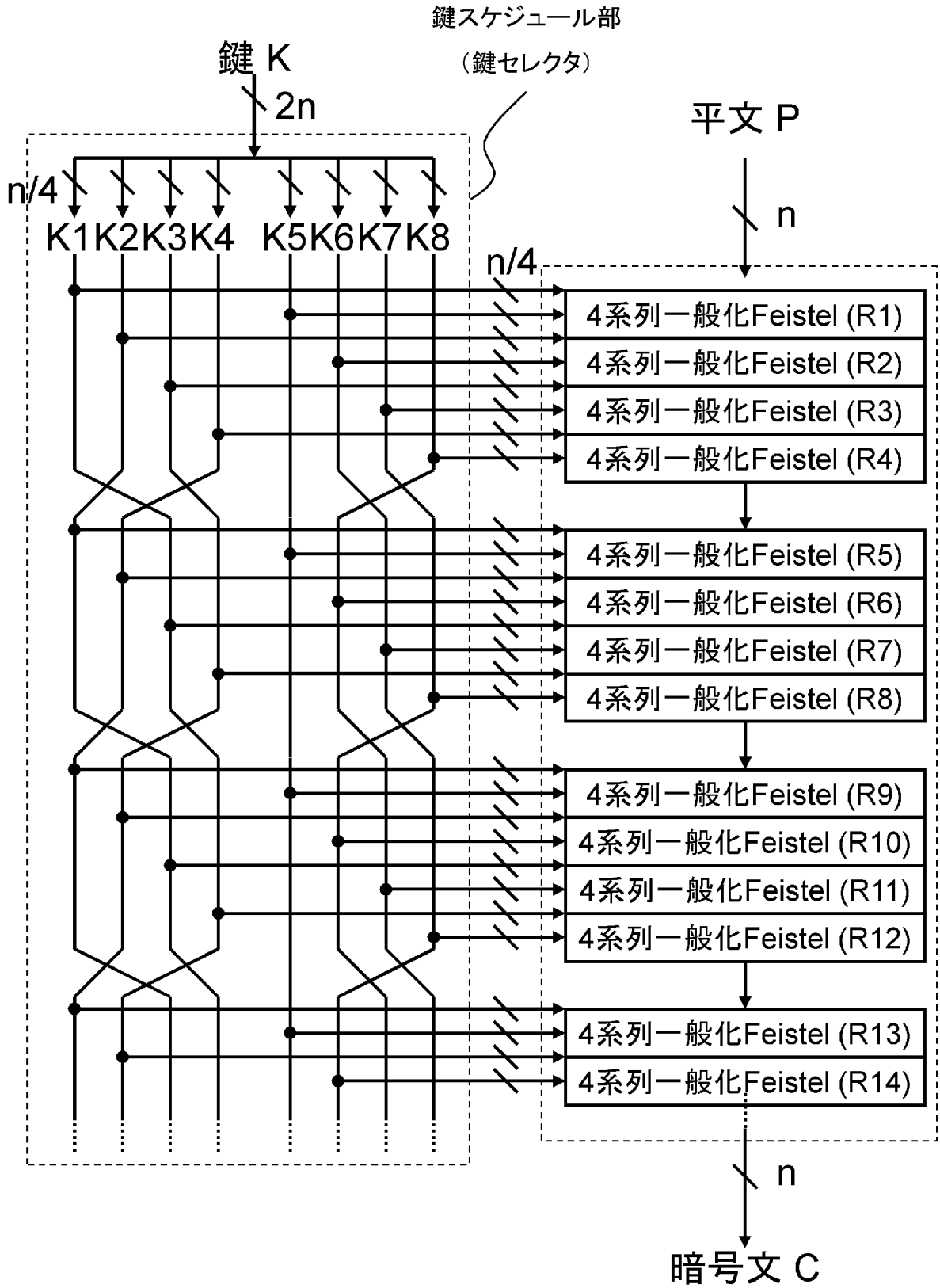
[図23]


鍵 K
4n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/2
K1K2K3K4K5K6K7K8
n/2
Feistel (R1)
Feistel (R2)
Feistel (R3)
Feistel (R4)
Feistel (R5)
Feistel (R6)
Feistel (R7)
Feistel (R8)
Feistel (R9)
Feistel (R10)
Feistel (R11)
Feistel (R12)
Feistel (R13)
Feistel (R14)
Feistel (R15)
Feistel (R16)
n
暗号文 C

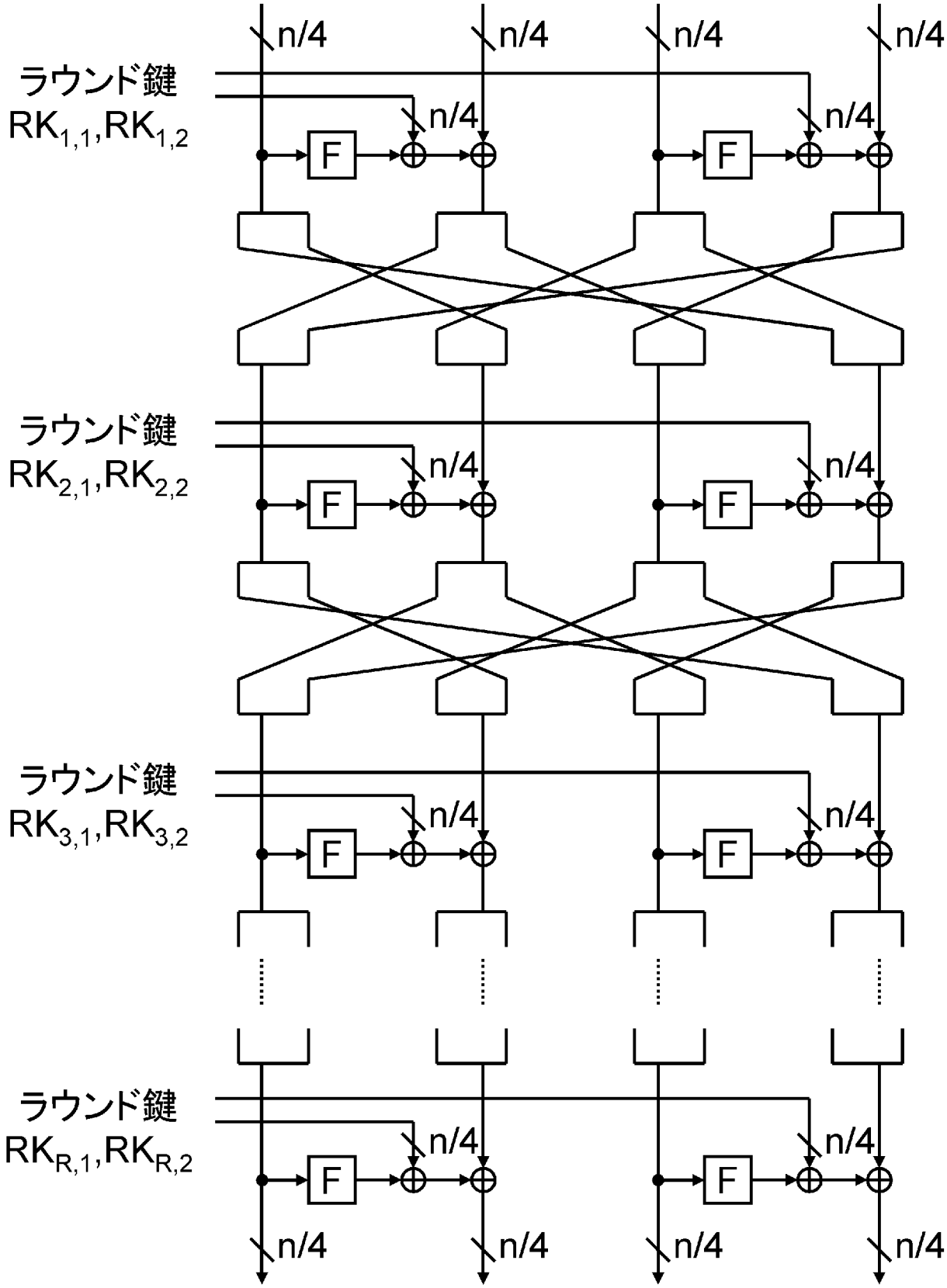
[図24]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n/4
K1K2K3K4K5K6K7K8
n/4
n
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
4系列一般化Feistel (R13)
4系列一般化Feistel (R14)
n
暗号文 C

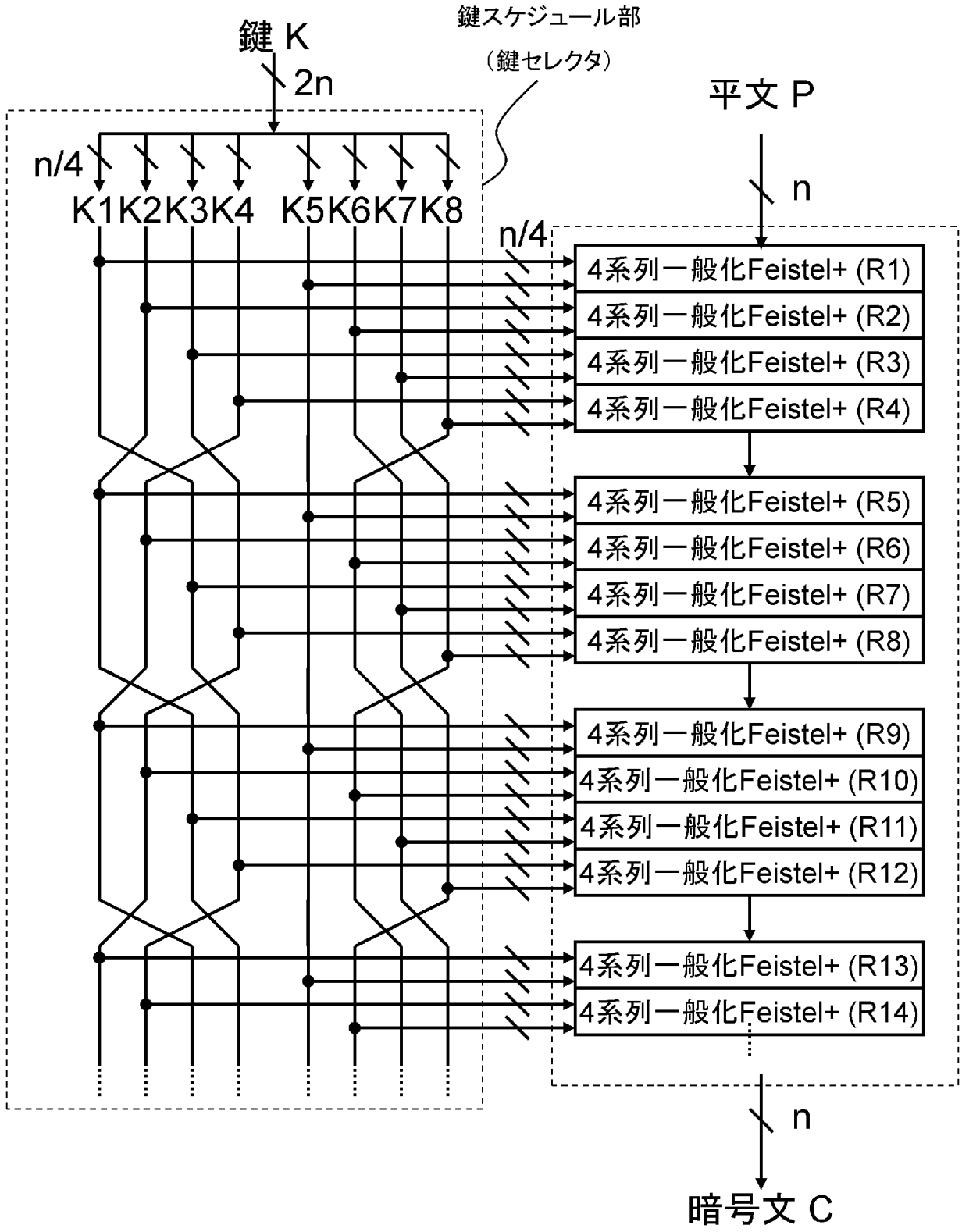
[図25]


鍵スケジュール部
（鍵セレクタ）
鍵 K
2n
n/4
K1K2K3K4 K5K6K7K8
平文 P
n
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
4系列一般化Feistel (R13)
4系列一般化Feistel (R14)
n
暗号文 C

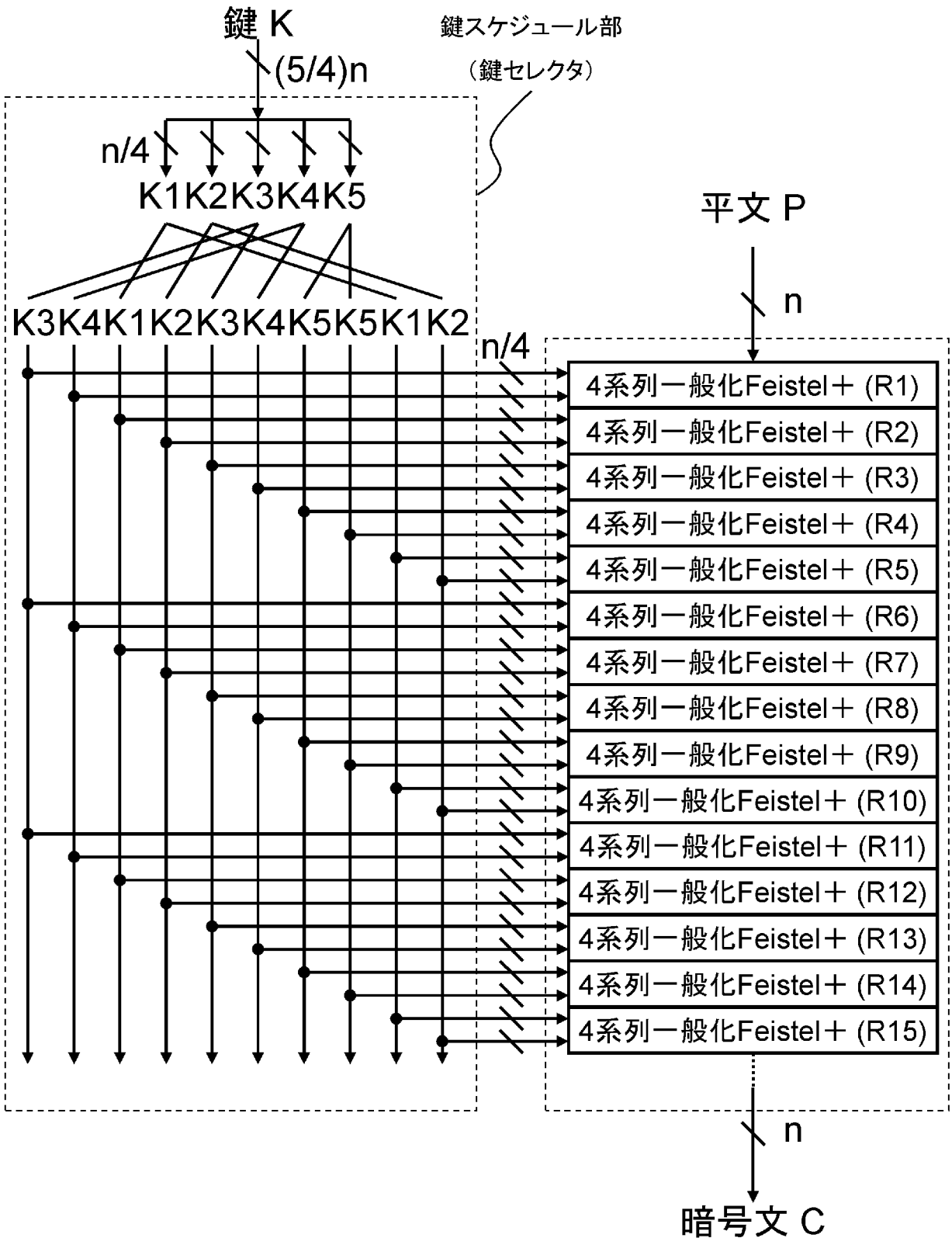
[図26]


n/4
n/4
n/4
n/4
ラウンド鍵
$RK_{1,1}, RK_{1,2}$
F
n/4
n/4
F
ラウンド鍵
$RK_{2,1}, RK_{2,2}$
F
n/4
n/4
F
ラウンド鍵
$RK_{3,1}, RK_{3,2}$
F
n/4
n/4
F
ラウンド鍵
$RK_{R,1}, RK_{R,2}$
F
n/4
n/4
F
n/4
n/4
n/4
n/4

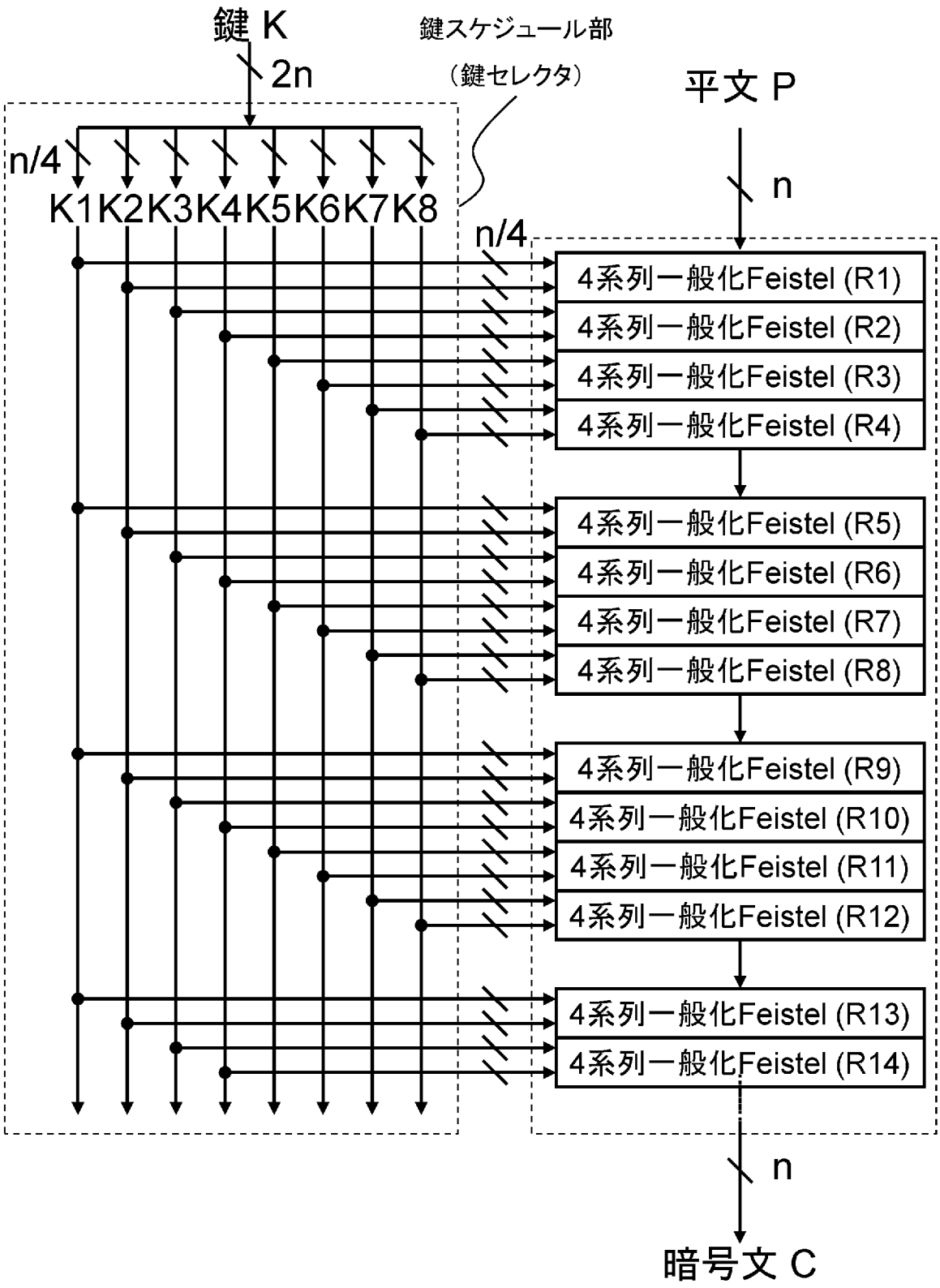
[図27]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4 K5K6K7K8
n/4
4系列一般化Feistel+ (R1)
4系列一般化Feistel+ (R2)
4系列一般化Feistel+ (R3)
4系列一般化Feistel+ (R4)
4系列一般化Feistel+ (R5)
4系列一般化Feistel+ (R6)
4系列一般化Feistel+ (R7)
4系列一般化Feistel+ (R8)
4系列一般化Feistel+ (R9)
4系列一般化Feistel+ (R10)
4系列一般化Feistel+ (R11)
4系列一般化Feistel+ (R12)
4系列一般化Feistel+ (R13)
4系列一般化Feistel+ (R14)
n
暗号文 C

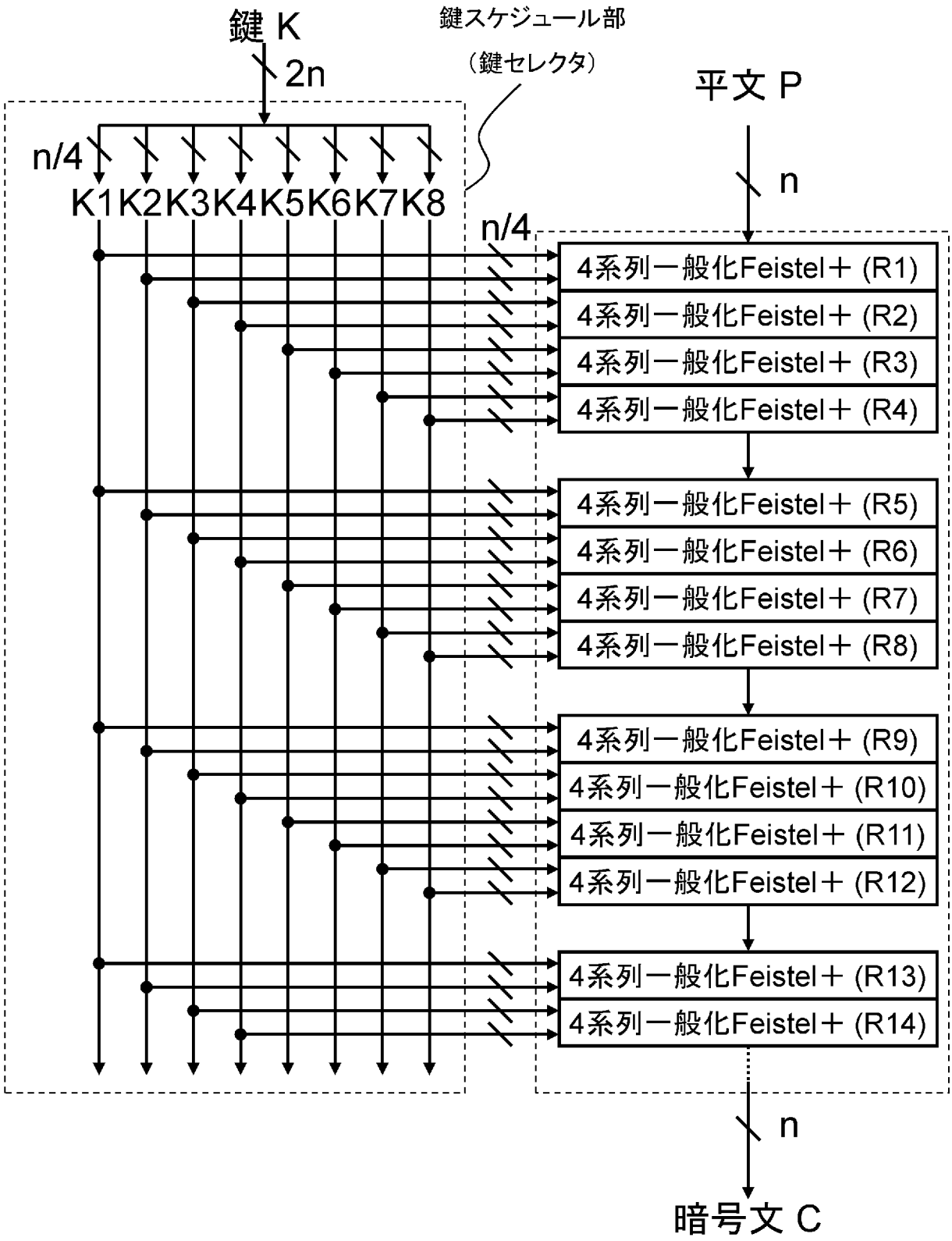
[図28]


鍵 K
(5/4)n
鍵スケジュール部
（鍵セレクタ）
n/4
K1K2K3K4K5
K3K4K1K2K3K4K5K5K1K2
n/4
平文 P
n
4系列一般化Feistel＋ (R1)
4系列一般化Feistel＋ (R2)
4系列一般化Feistel＋ (R3)
4系列一般化Feistel＋ (R4)
4系列一般化Feistel＋ (R5)
4系列一般化Feistel＋ (R6)
4系列一般化Feistel＋ (R7)
4系列一般化Feistel＋ (R8)
4系列一般化Feistel＋ (R9)
4系列一般化Feistel＋ (R10)
4系列一般化Feistel＋ (R11)
4系列一般化Feistel＋ (R12)
4系列一般化Feistel＋ (R13)
4系列一般化Feistel＋ (R14)
4系列一般化Feistel＋ (R15)
n
暗号文 C

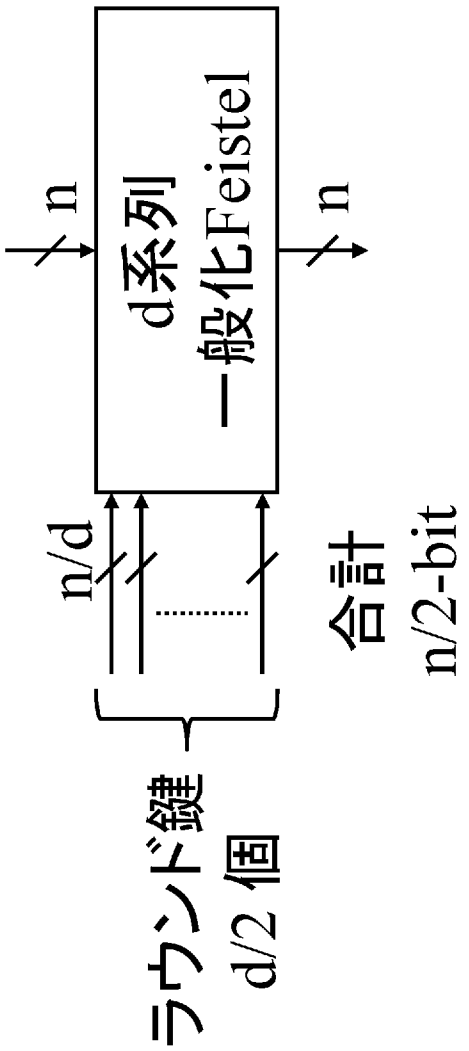
[図29]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5K6K7K8
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
4系列一般化Feistel (R13)
4系列一般化Feistel (R14)
n
暗号文 C

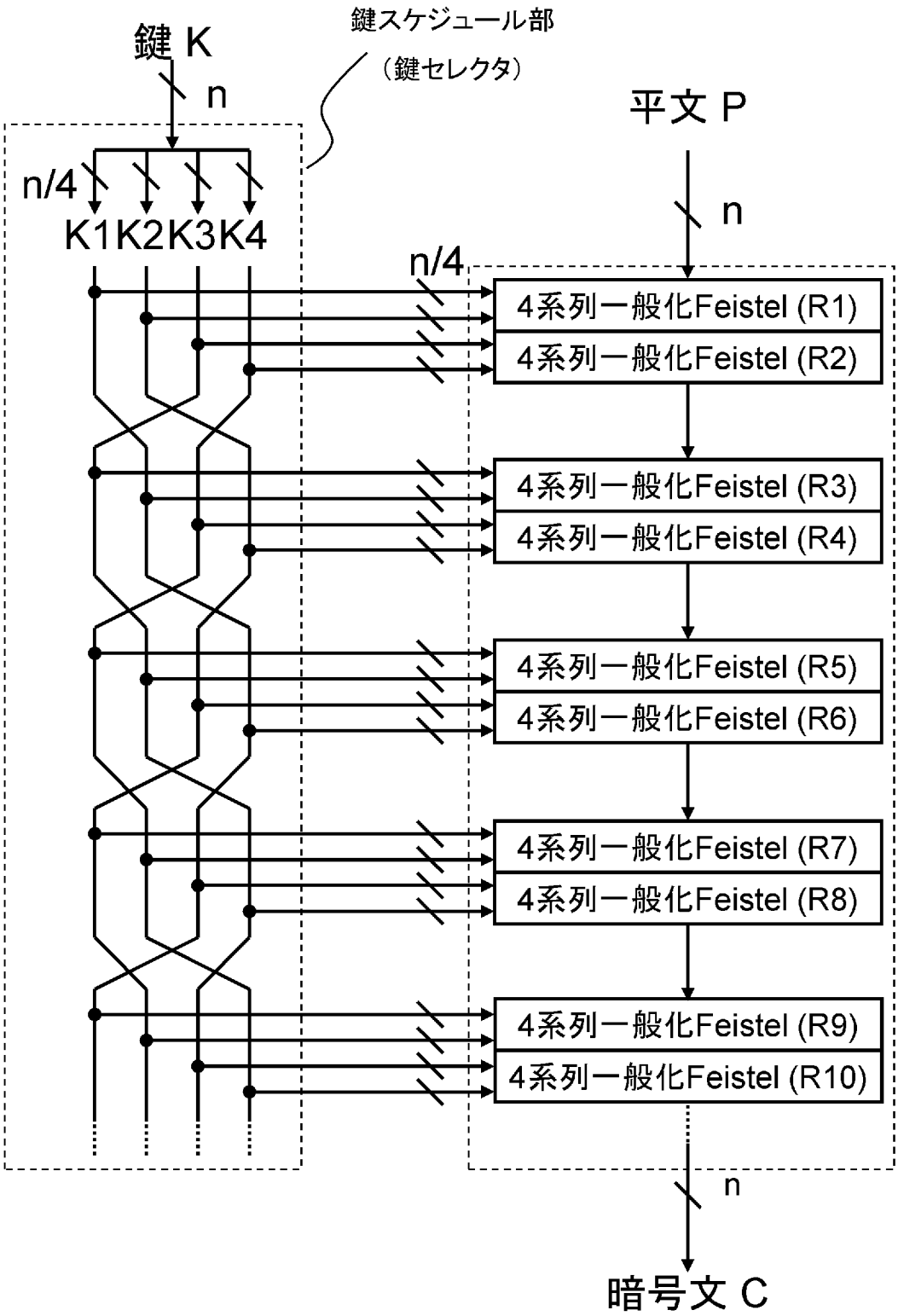
[図30]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5K6K7K8
n/4
4系列一般化Feistel＋ (R1)
4系列一般化Feistel＋ (R2)
4系列一般化Feistel＋ (R3)
4系列一般化Feistel＋ (R4)
4系列一般化Feistel＋ (R5)
4系列一般化Feistel＋ (R6)
4系列一般化Feistel＋ (R7)
4系列一般化Feistel＋ (R8)
4系列一般化Feistel＋ (R9)
4系列一般化Feistel＋ (R10)
4系列一般化Feistel＋ (R11)
4系列一般化Feistel＋ (R12)
4系列一般化Feistel＋ (R13)
4系列一般化Feistel＋ (R14)
n
暗号文 C

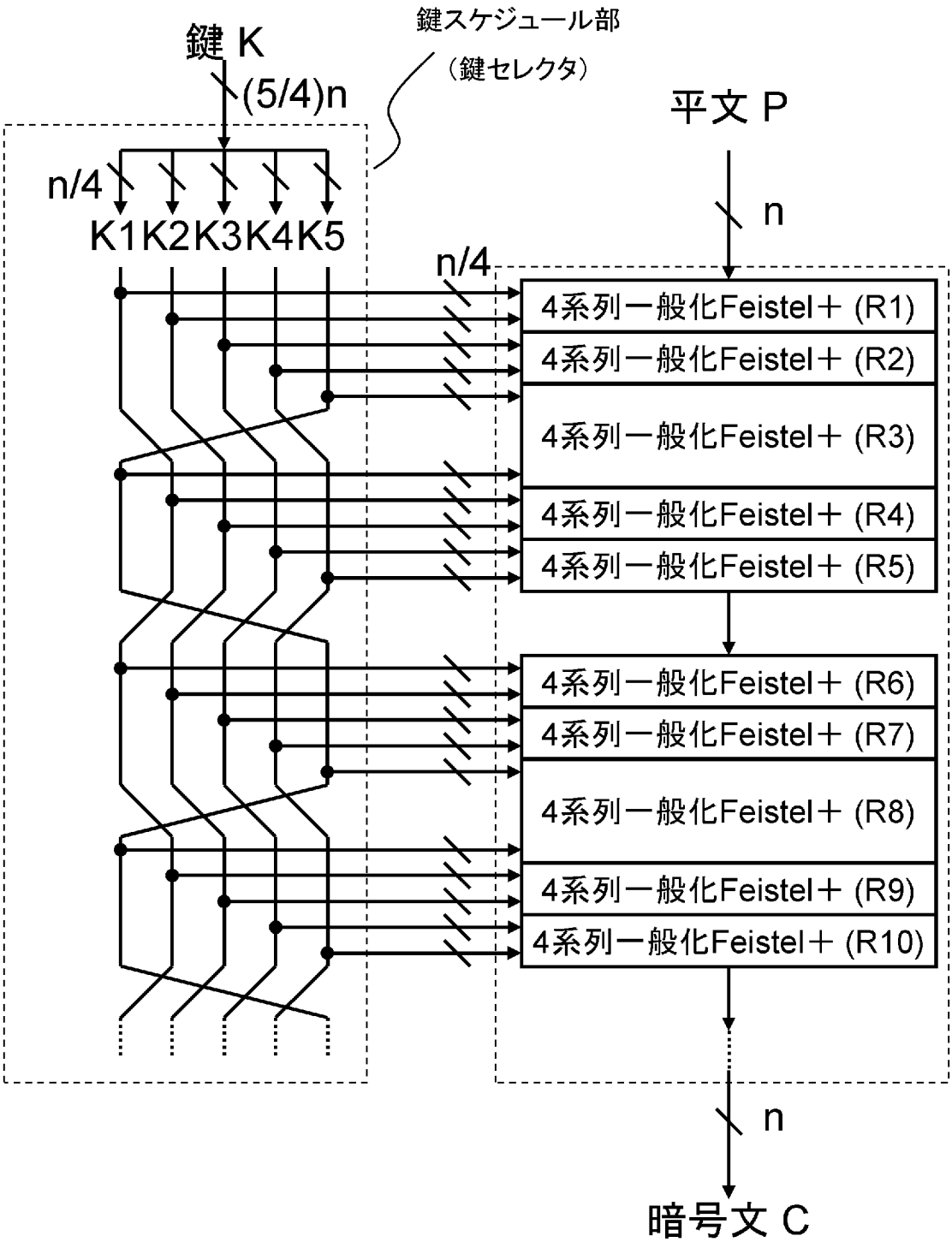
[図31]



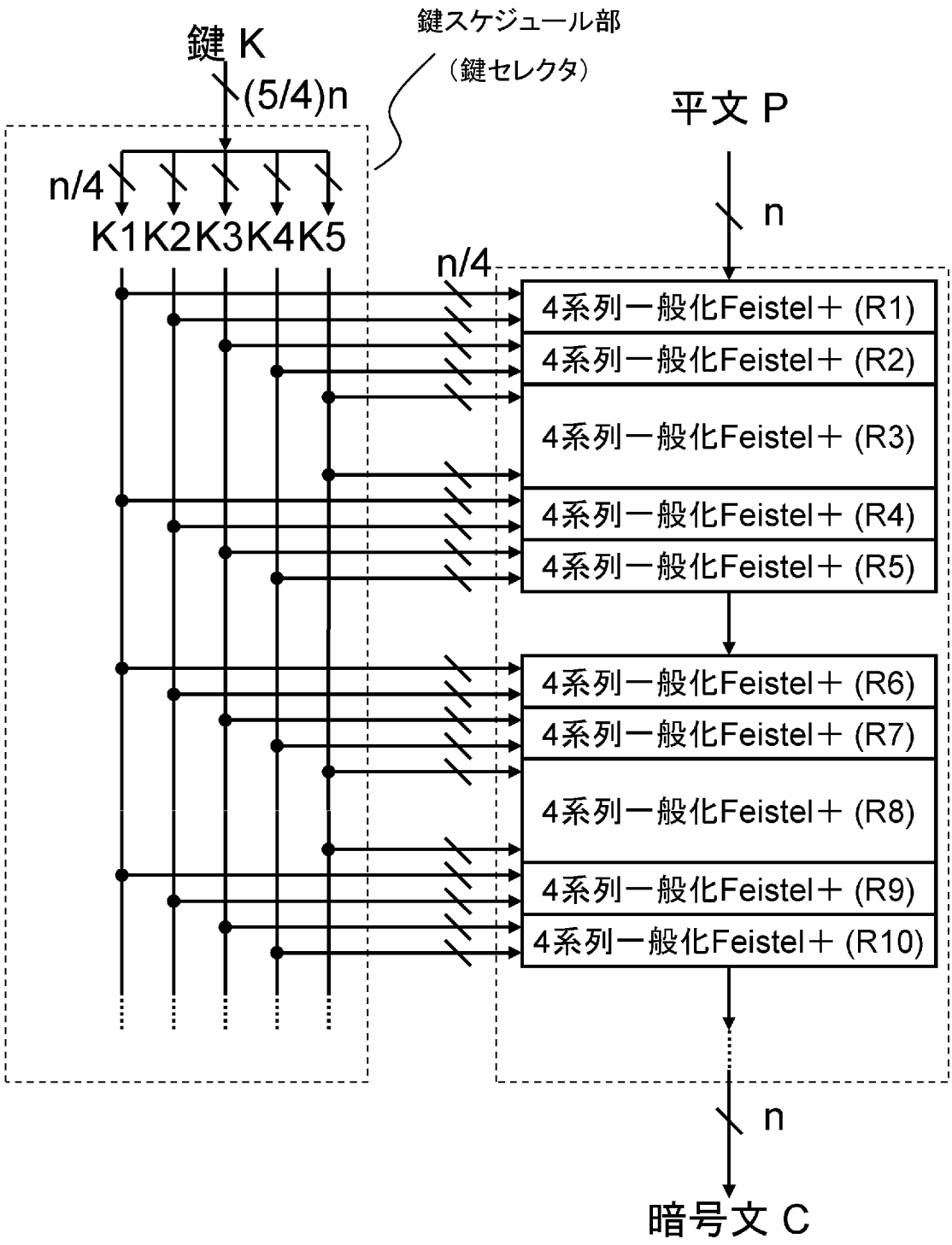
[図32]


鍵 K
n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
n
暗号文 C

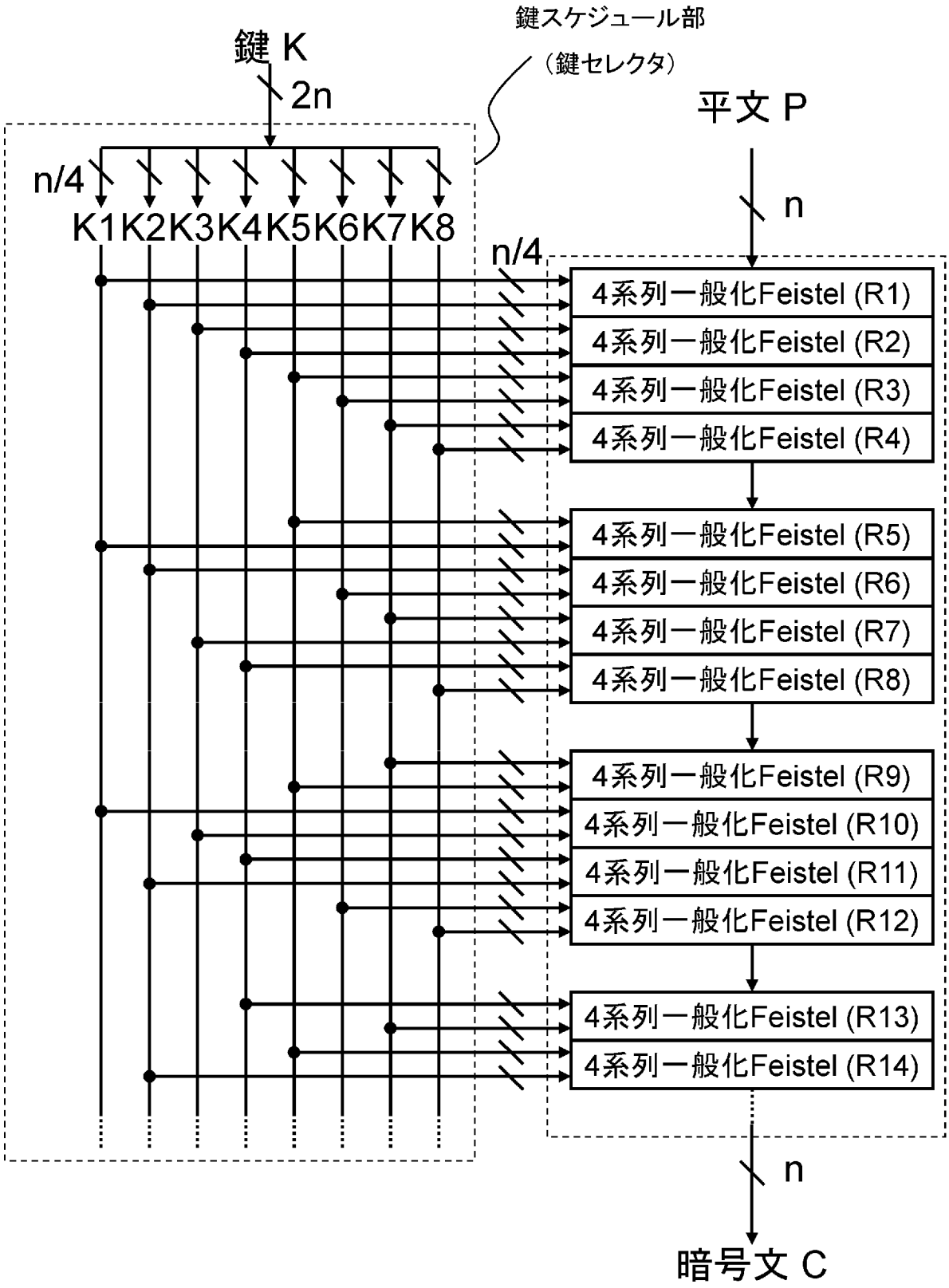
[図33]


鍵 K
(5/4)n
鍵スケジュール部
（鍵セレクタ）
n/4
K1K2K3K4K5
平文 P
n
n/4
4系列一般化Feistel＋ (R1)
4系列一般化Feistel＋ (R2)
4系列一般化Feistel＋ (R3)
4系列一般化Feistel＋ (R4)
4系列一般化Feistel＋ (R5)
4系列一般化Feistel＋ (R6)
4系列一般化Feistel＋ (R7)
4系列一般化Feistel＋ (R8)
4系列一般化Feistel＋ (R9)
4系列一般化Feistel＋ (R10)
n
暗号文 C

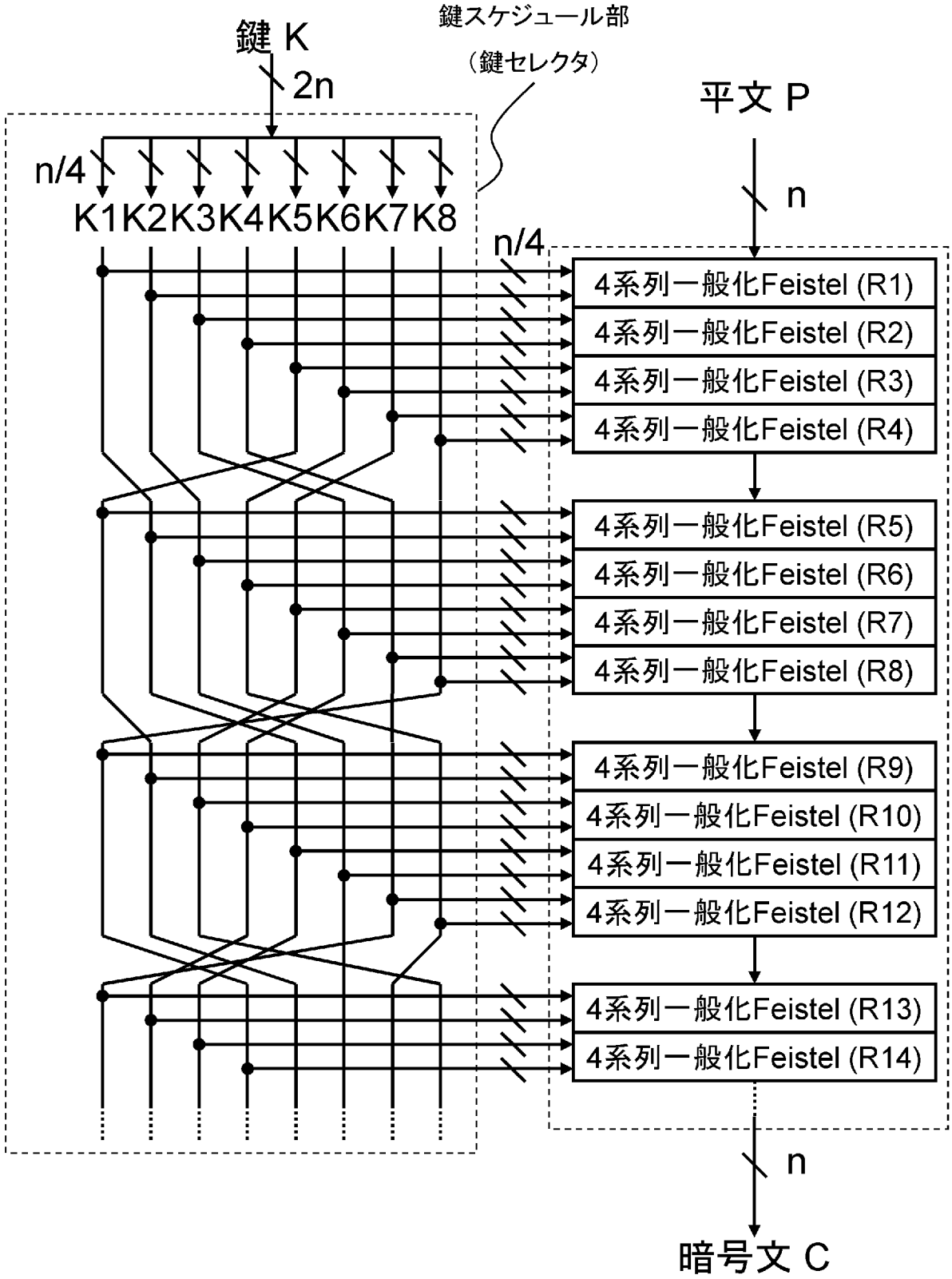
[図34]


鍵 K
(5/4)n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5
n/4
4系列一般化Feistel＋ (R1)
4系列一般化Feistel＋ (R2)
4系列一般化Feistel＋ (R3)
4系列一般化Feistel＋ (R4)
4系列一般化Feistel＋ (R5)
4系列一般化Feistel＋ (R6)
4系列一般化Feistel＋ (R7)
4系列一般化Feistel＋ (R8)
4系列一般化Feistel＋ (R9)
4系列一般化Feistel＋ (R10)
n
暗号文 C

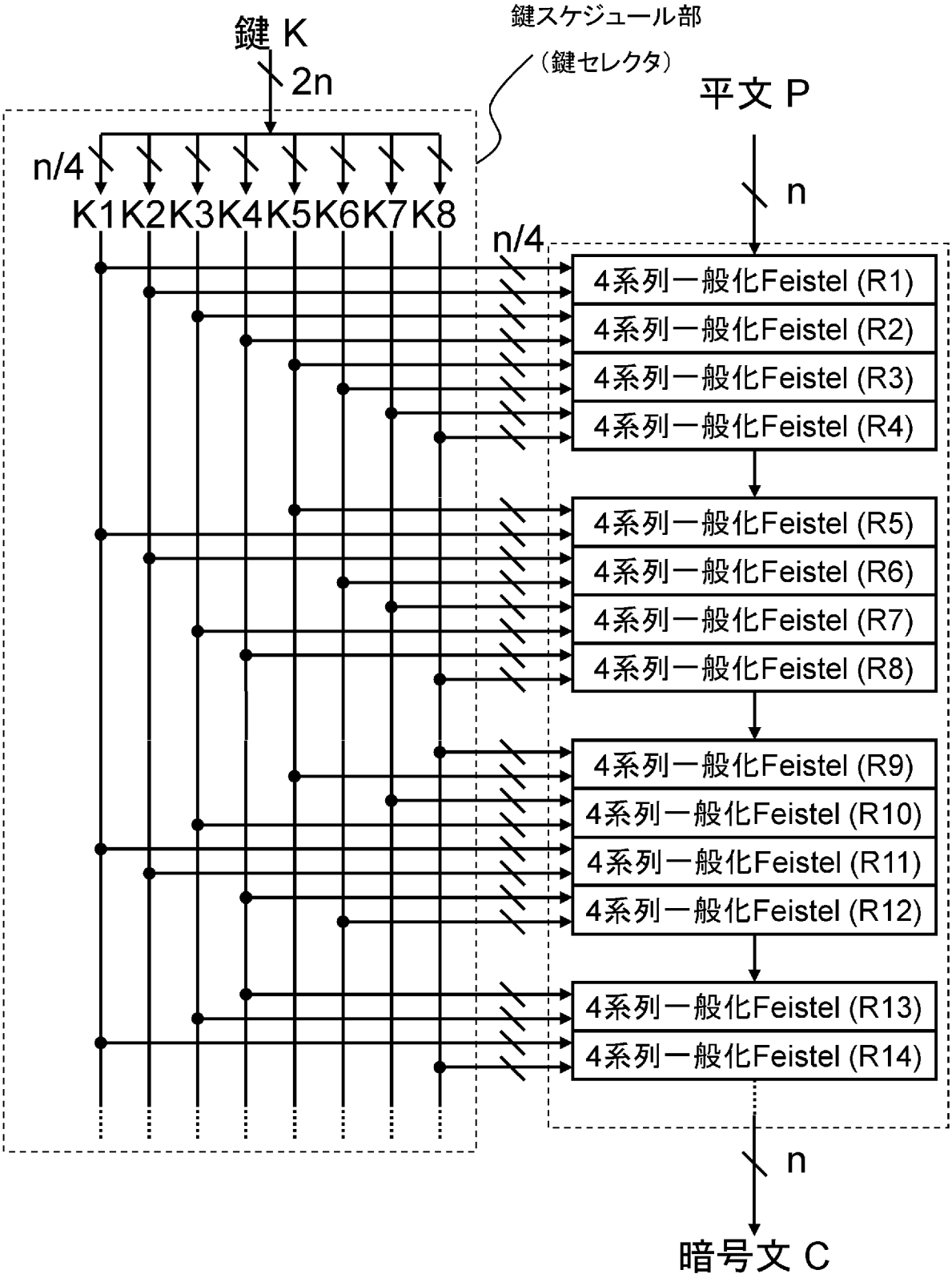
[図35]


鍵スケジュール部
（鍵セレクタ）
鍵 K
2n
n/4
K1K2K3K4K5K6K7K8
平文 P
n
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
4系列一般化Feistel (R13)
4系列一般化Feistel (R14)
n
暗号文 C

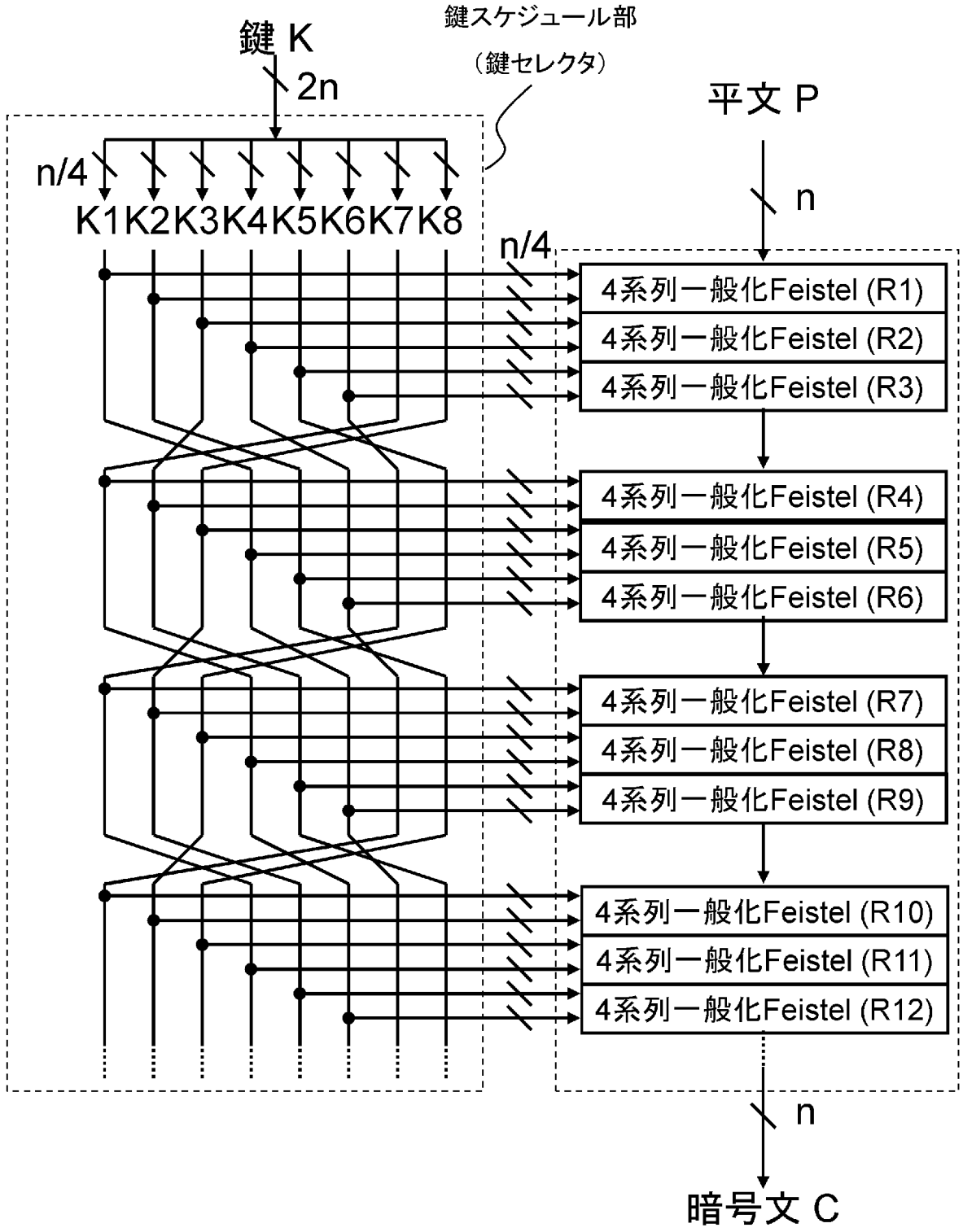
[図36]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5K6K7K8
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
4系列一般化Feistel (R13)
4系列一般化Feistel (R14)
n
暗号文 C

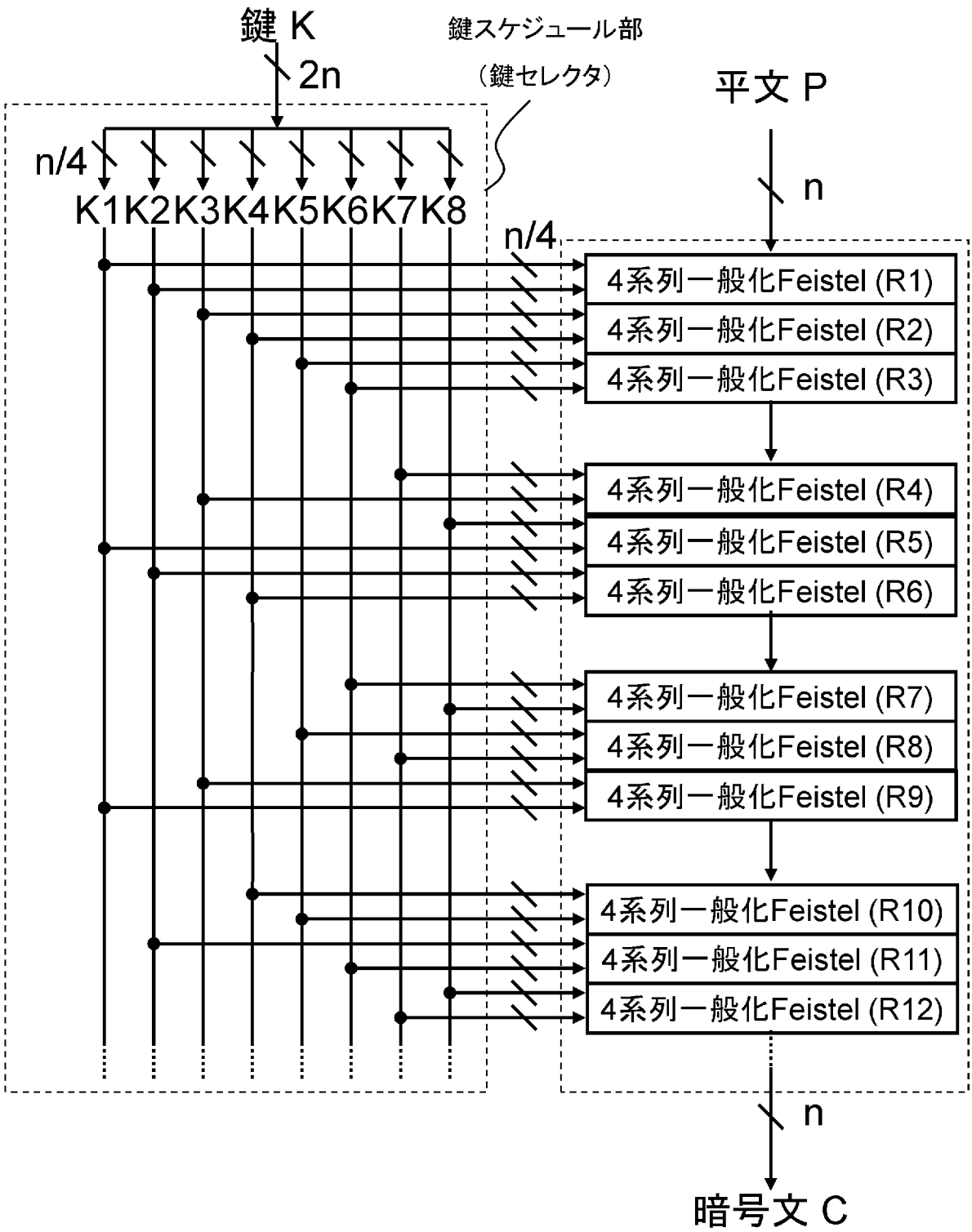
[図37]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5K6K7K8
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
4系列一般化Feistel (R13)
4系列一般化Feistel (R14)
n
暗号文 C

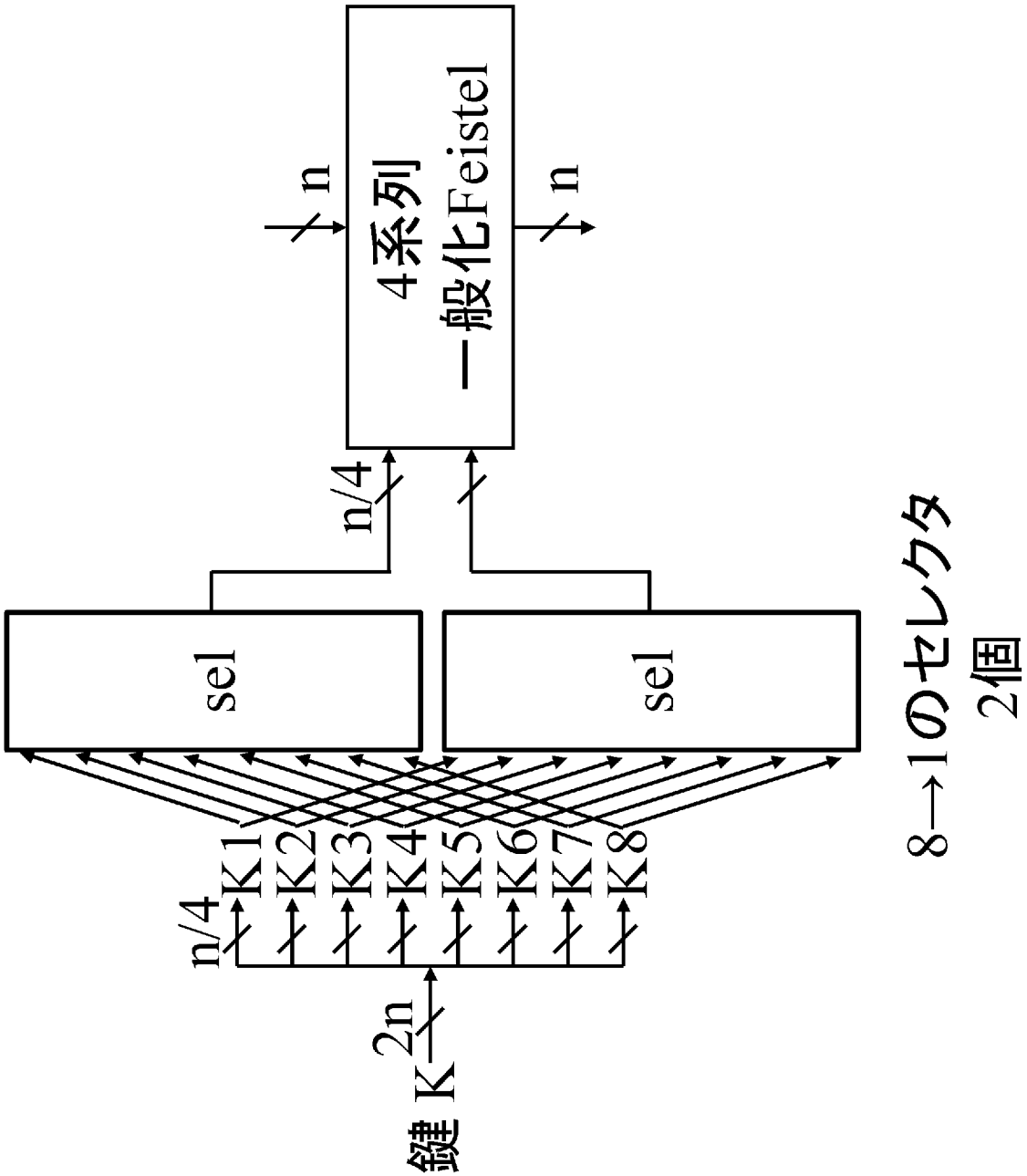
[図38]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5K6K7K8
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
n
暗号文 C

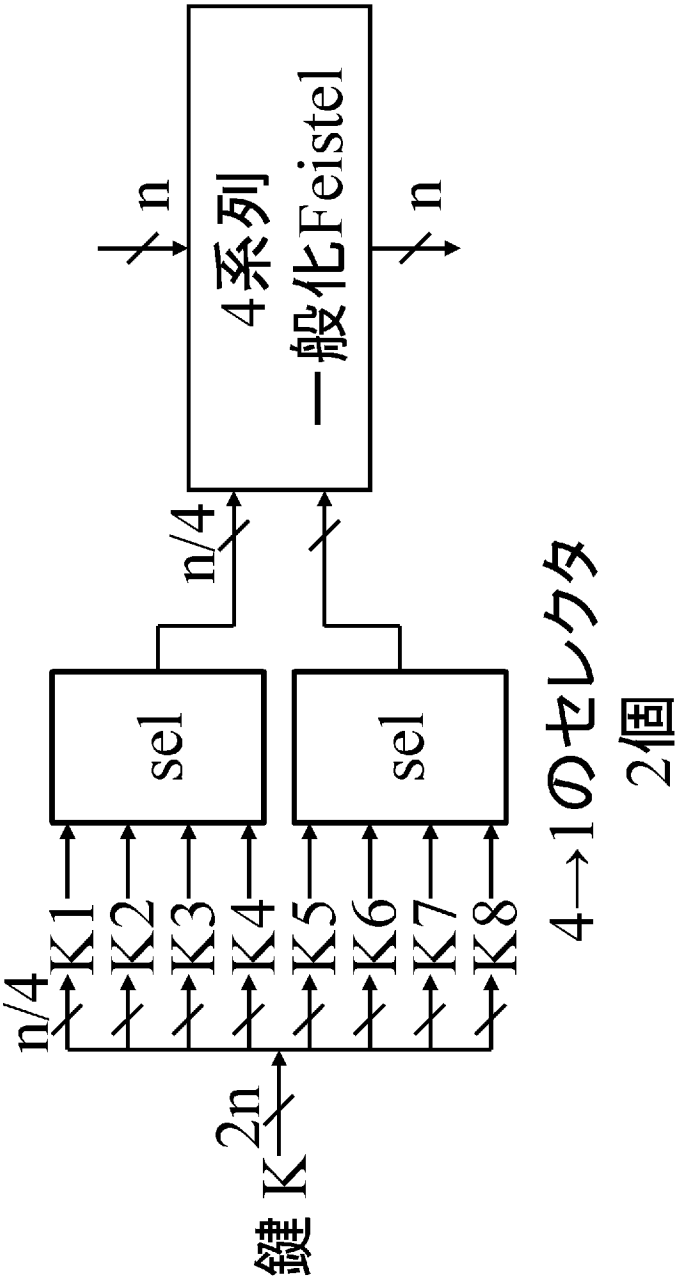
[図39]


鍵 K
2n
鍵スケジュール部
（鍵セレクタ）
平文 P
n
n/4
K1K2K3K4K5K6K7K8
n/4
4系列一般化Feistel (R1)
4系列一般化Feistel (R2)
4系列一般化Feistel (R3)
4系列一般化Feistel (R4)
4系列一般化Feistel (R5)
4系列一般化Feistel (R6)
4系列一般化Feistel (R7)
4系列一般化Feistel (R8)
4系列一般化Feistel (R9)
4系列一般化Feistel (R10)
4系列一般化Feistel (R11)
4系列一般化Feistel (R12)
n
暗号文 C

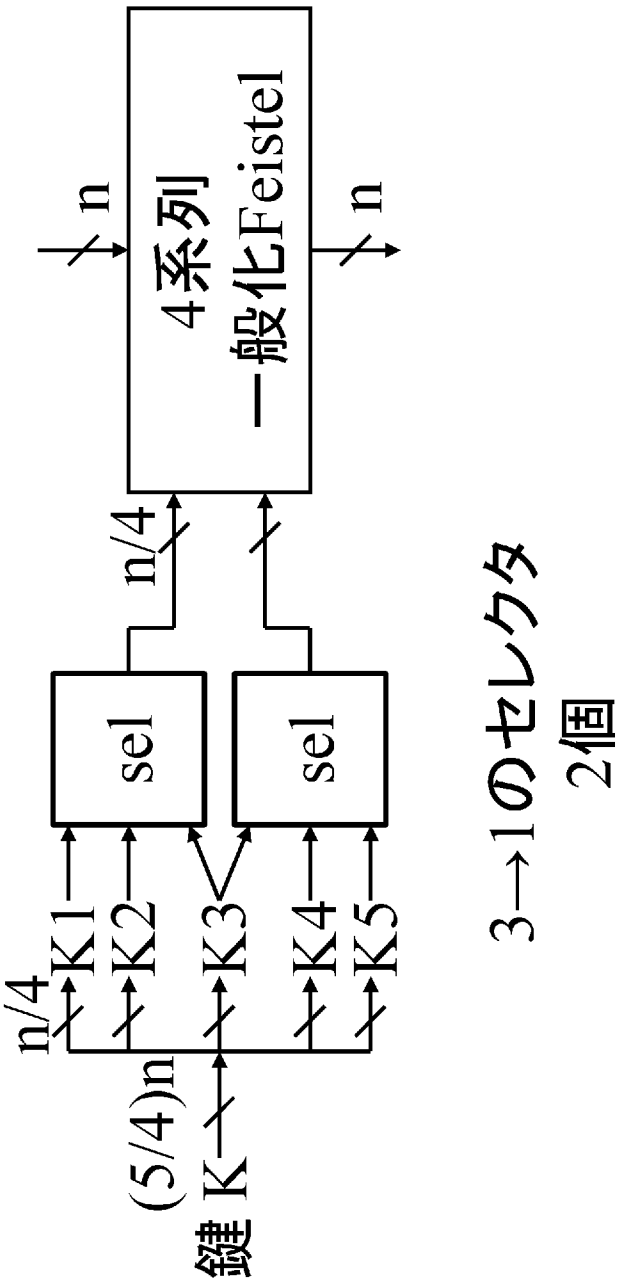
[図40]


n
4系列
一般化Feistel
n
n/4
sel
sel
8→1のセレクタ
2個
n/4
K1
K2
K3
K4
K5
K6
K7
K8
2n
鍵 K

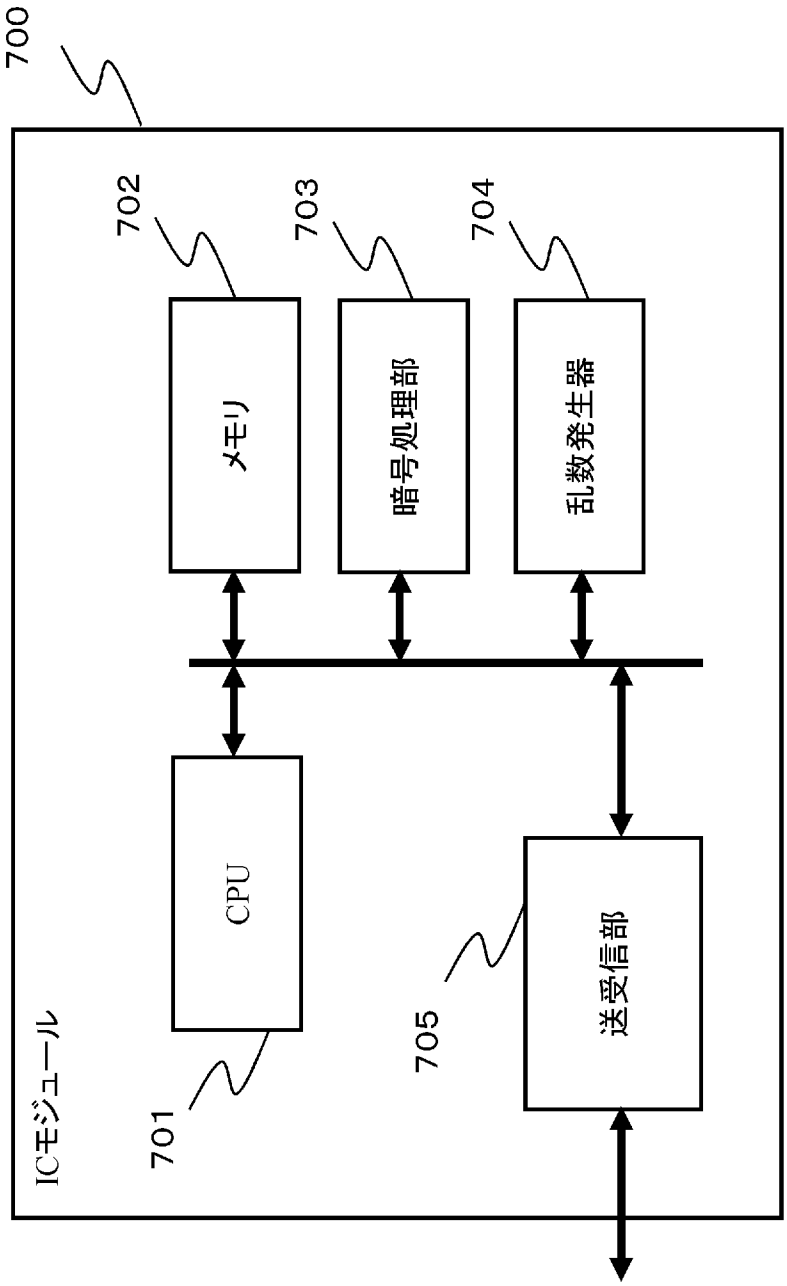
[図41]


n
4系列
一般化Feistel
n
n/4
sel
sel
4→1のセレクタ
2個
K1
K2
K3
K4
K5
K6
K7
K8
n/4
2n
鍵 K

[図42]



[図43]



# INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2012/053933

A. CLASSIFICATION OF SUBJECT MATTER
*H04L9/06*(2006.01)i, *G09C1/00*(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L9/06, G09C1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

| | | | |
|---|---|---|---|
| Jitsuyo Shinan Koho | 1922-1996 | Jitsuyo Shinan Toroku Koho | 1996-2012 |
| Kokai Jitsuyo Shinan Koho | 1971-2012 | Toroku Jitsuyo Shinan Koho | 1994-2012 |

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JSTPlus/JMEDPlus/JST7580(JDreamII), cipher, key schedule

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | Ko, Y. et al., Related Key Differential Attacks on 27 Rounds of XTEA and Full-Round GOST, Lecture Notes in Computer Science, Vol.3017, 2004.07.28, p.299-316, especially 3.1 Description of XTEA | 1-12 |
| P,X | Shibutani, K. et al., Piccolo: An Ultra-Lightweight Blockcipher, Lecture Notes in Computer Science, Vol.6917, 2011.09.27, p.342-357, especially 2.3 Key Scheduling Part, 5.1 Optimization in Key Scheduling Part | 1-12 |

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 21 March, 2012 (21.03.12) | 03 April, 2012 (03.04.12) |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| Japanese Patent Office | |
| Facsimile No. | Telephone No. |

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２０１２／０５３９３３

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　H04L9/06(2006.01)i, G09C1/00(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　H04L9/06, G09C1/00

最小限資料以外の資料で調査を行った分野に含まれるもの

| | |
|---|---|
| 日本国実用新案公報 | １９２２－１９９６年 |
| 日本国公開実用新案公報 | １９７１－２０１２年 |
| 日本国実用新案登録公報 | １９９６－２０１２年 |
| 日本国登録実用新案公報 | １９９４－２０１２年 |

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

JSTPlus/JMEDPlus/JST7580(JDreamII) cipher, key schedule

Ｃ．　関連すると認められる文献

| 引用文献のカテゴリー＊ | 引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示 | 関連する請求項の番号 |
|---|---|---|
| X | Ko, Y. et al., Related Key Differential Attacks on 27 Rounds of XTEA and Full-Round GOST, Lecture Notes in Computer Science, Vol.3017, 2004.07.28, p.299-316, especially 3.1 Description of XTEA | 1-12 |
| P, X | Shibutani, K. et al., Piccolo: An Ultra-Lightweight Blockcipher, Lecture Notes in Computer Science, Vol.6917, 2011.09.27, p.342-357, especially 2.3 Key Scheduling Part, 5.1 Optimization in Key Scheduling Part | 1-12 |

☐　Ｃ欄の続きにも文献が列挙されている。

☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

| | |
|---|---|
| 国際調査を完了した日<br>２１．０３．２０１２ | 国際調査報告の発送日<br>０３．０４．２０１２ |
| 国際調査機関の名称及びあて先<br>日本国特許庁（ＩＳＡ／ＪＰ）<br>郵便番号１００－８９１５<br>東京都千代田区霞が関三丁目４番３号 | 特許庁審査官（権限のある職員）　５Ｓ　９３６４<br>中里　裕正<br>電話番号　０３－３５８１－１１０１　内線　３５４６ |