(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107896149 A
(43)申请公布日 2018.04.10

(21)申请号 201810000488.6

(22)申请日 2018.01.02

(71)申请人 南京航空航天大学
地址 211106 江苏省南京市江宁区将军大道29号

(72)发明人 苏盛辉 郑建华 王箭

(51)Int.Cl.
H04L 9/08(2006.01)
H04L 9/06(2006.01)

权利要求书4页 说明书8页

(54)发明名称

基于三个群运算的128位对称加密方法

(57)摘要

基于三个群运算的128位对称加密方法，属于密码技术和计算机技术领域；包括密钥生成、加密和解密三个部分；发送方随机产生一个256位的初始密钥，使用密钥生成部分从它得到各由72个子密钥组成的加密密钥和解密密钥，并通过安全途径把解密密钥传输给接收方；发送方使用加密密钥和加密部分把128位的明文转换成密文(加密)；接收方使用解密密钥和解密部分把128位的密文还原成明文(解密)；该方法具有安全性高、计算速度快、技术可以公开、适用性强等特点，可用于计算机和通信网络中任何文件、数据的保密存储与传输。

CN 107896149 A

1.基于三个群运算的128位对称加密方法，由密钥生成、加密和解密三个部分组成，密钥生成部分供发送方利用随机初始密钥产生加密密钥和解密密钥，并负责把解密密钥通过安全途径传输给接收方，加密部分供发送方利用加密密钥把明文分组转化为密文分组，解密部分供接收方利用解密密钥把密文分组还原成明文分组，在下文中，符号←表示把右边的常数值或表达式值赋给左边的变量，“位”表示一个二进制位（也称为比特，其值为0或1），%代表模运算，⊕代表两个16位运算数的模2加法（即比特异或），[+]代表两个16位运算数的模2^{16}加法，⊙代表两个16位运算数的模2^{16}+1乘法，$K_i^{(j)}$和$K'^{(j)}_i$分别代表加密和解密第j轮第i个子密钥，$-K_i^{(j)}$表示$K_i^{(j)}$模2^{16}的加法逆元，$(K_i^{(j)})^{-1}$表示$K_i^{(j)}$模2^{16}+1的乘法逆元，特别，使用16位全零元素代表2^{16}以及2^{16}的乘法逆元，并且，任何一个元素和16位全零元素做模2^{16}+1乘法运算时，16位全零元素应被视作2^{16}，任何一个元素和16位全零元素做模2^{16}加法运算时，16位全零元素仍被视作0，本方法的特征在于

●密钥生成部分采用了下列步骤：

输入：一个256位的随机初始密钥K

(1) 置j←0；

(2) 把K最左边128位划分为8个16位的子密钥，分别赋给$K_0^{(j)}$，$K_1^{(j)}$，...，$K_7^{(j)}$；

(3) 把K循环左移25位；

(4) 令j←j+1；

(5) 如果j≤8，则转至(2)，否则，下一步；

(6) 令$K'^{(0)}_0, K'^{(1)}_0, \ldots, K'^{(8)}_0 \leftarrow -K_0^{(8)}, -K_0^{(7)}, \ldots, -K_0^{(0)}$，

令$K'^{(0)}_3, K'^{(1)}_3, \ldots, K'^{(8)}_3 \leftarrow -K_3^{(8)}, -K_3^{(7)}, \ldots, -K_3^{(0)}$，

令$K'^{(0)}_4, K'^{(1)}_4, \ldots, K'^{(8)}_4 \leftarrow (K_4^{(8)})^{-1}, (K_4^{(7)})^{-1}, \ldots, (K_4^{(0)})^{-1}$，

令$K'^{(0)}_7, K'^{(1)}_7, \ldots, K'^{(8)}_7 \leftarrow (K_7^{(8)})^{-1}, (K_7^{(7)})^{-1}, \ldots, (K_7^{(0)})^{-1}$；

(7) 令$K'^{(0)}_1 \leftarrow (K_1^{(8)})^{-1}$，$K'^{(8)}_1 \leftarrow (K_1^{(0)})^{-1}$，$K'^{(0)}_2 \leftarrow (K_2^{(8)})^{-1}$，$K'^{(8)}_2 \leftarrow (K_2^{(0)})^{-1}$，

令$K'^{(0)}_5 \leftarrow -K_5^{(8)}$，$K'^{(8)}_5 \leftarrow -K_5^{(0)}$，$K'^{(0)}_6 \leftarrow -K_6^{(8)}$，$K'^{(8)}_6 \leftarrow -K_6^{(0)}$；

(8) 令$K'^{(1)}_1, K'^{(2)}_1, \ldots, K'^{(7)}_1 \leftarrow (K_2^{(7)})^{-1}, (K_2^{(6)})^{-1}, \ldots, (K_2^{(1)})^{-1}$，

令$K'^{(1)}_2, K'^{(2)}_2, \ldots, K'^{(7)}_2 \leftarrow (K_1^{(7)})^{-1}, (K_1^{(6)})^{-1}, \ldots, (K_1^{(1)})^{-1}$，

令$K'^{(1)}_5, K'^{(2)}_5, \ldots, K'^{(7)}_5 \leftarrow -K_6^{(7)}, -K_6^{(6)}, \ldots, -K_6^{(1)}$，

令$K'^{(1)}_6, K'^{(2)}_6, \ldots, K'^{(7)}_6 \leftarrow -K_5^{(7)}, -K_5^{(6)}, \ldots, -K_5^{(1)}$；

输出：加密密钥（由72个子密钥组成）

$$K_0^{(0)} \quad K_1^{(0)} \quad K_2^{(0)} \quad K_3^{(0)} \quad K_4^{(0)} \quad K_5^{(0)} \quad K_6^{(0)} \quad K_7^{(0)}$$

$$K_0^{(1)} \quad K_1^{(1)} \quad K_2^{(1)} \quad K_3^{(1)} \quad K_4^{(1)} \quad K_5^{(1)} \quad K_6^{(1)} \quad K_7^{(1)}$$

$$K_0^{(2)} \quad K_1^{(2)} \quad K_2^{(2)} \quad K_3^{(2)} \quad K_4^{(2)} \quad K_5^{(2)} \quad K_6^{(2)} \quad K_7^{(2)}$$

$K_0^{(3)}$ $K_1^{(3)}$ $K_2^{(3)}$ $K_3^{(3)}$ $K_4^{(3)}$ $K_5^{(3)}$ $K_6^{(3)}$ $K_7^{(3)}$

$K_0^{(4)}$ $K_1^{(4)}$ $K_2^{(4)}$ $K_3^{(4)}$ $K_4^{(4)}$ $K_5^{(4)}$ $K_6^{(4)}$ $K_7^{(4)}$

$K_0^{(5)}$ $K_1^{(5)}$ $K_2^{(5)}$ $K_3^{(5)}$ $K_4^{(5)}$ $K_5^{(5)}$ $K_6^{(5)}$ $K_7^{(5)}$

$K_0^{(6)}$ $K_1^{(6)}$ $K_2^{(6)}$ $K_3^{(6)}$ $K_4^{(6)}$ $K_5^{(6)}$ $K_6^{(6)}$ $K_7^{(6)}$

$K_0^{(7)}$ $K_1^{(7)}$ $K_2^{(7)}$ $K_3^{(7)}$ $K_4^{(7)}$ $K_5^{(7)}$ $K_6^{(7)}$ $K_7^{(7)}$

$K_0^{(8)}$ $K_1^{(8)}$ $K_2^{(8)}$ $K_3^{(8)}$ $K_4^{(8)}$ $K_5^{(8)}$ $K_6^{(8)}$ $K_7^{(8)}$;

解密密钥(由72个子密钥组成)

$K'_0{}^{(0)}$ $K'_1{}^{(0)}$ $K'_2{}^{(0)}$ $K'_3{}^{(0)}$ $K'_4{}^{(0)}$ $K'_5{}^{(0)}$ $K'_6{}^{(0)}$ $K'_7{}^{(0)}$

$K'_0{}^{(1)}$ $K'_1{}^{(1)}$ $K'_2{}^{(1)}$ $K'_3{}^{(1)}$ $K'_4{}^{(1)}$ $K'_5{}^{(1)}$ $K'_6{}^{(1)}$ $K'_7{}^{(1)}$

$K'_0{}^{(2)}$ $K'_1{}^{(2)}$ $K'_2{}^{(2)}$ $K'_3{}^{(2)}$ $K'_4{}^{(2)}$ $K'_5{}^{(2)}$ $K'_6{}^{(2)}$ $K'_7{}^{(2)}$

$K'_0{}^{(3)}$ $K'_1{}^{(3)}$ $K'_2{}^{(3)}$ $K'_3{}^{(3)}$ $K'_4{}^{(3)}$ $K'_5{}^{(3)}$ $K'_6{}^{(3)}$ $K'_7{}^{(3)}$

$K'_0{}^{(4)}$ $K'_1{}^{(4)}$ $K'_2{}^{(4)}$ $K'_3{}^{(4)}$ $K'_4{}^{(4)}$ $K'_5{}^{(4)}$ $K'_6{}^{(4)}$ $K'_7{}^{(4)}$

$K'_0{}^{(5)}$ $K'_1{}^{(5)}$ $K'_2{}^{(5)}$ $K'_3{}^{(5)}$ $K'_4{}^{(5)}$ $K'_5{}^{(5)}$ $K'_6{}^{(5)}$ $K'_7{}^{(5)}$

$K'_0{}^{(6)}$ $K'_1{}^{(6)}$ $K'_2{}^{(6)}$ $K'_3{}^{(6)}$ $K'_4{}^{(6)}$ $K'_5{}^{(6)}$ $K'_6{}^{(6)}$ $K'_7{}^{(6)}$

$K'_0{}^{(7)}$ $K'_1{}^{(7)}$ $K'_2{}^{(7)}$ $K'_3{}^{(7)}$ $K'_4{}^{(7)}$ $K'_5{}^{(7)}$ $K'_6{}^{(7)}$ $K'_7{}^{(7)}$

$K'_0{}^{(8)}$ $K'_1{}^{(8)}$ $K'_2{}^{(8)}$ $K'_3{}^{(8)}$ $K'_4{}^{(8)}$ $K'_5{}^{(8)}$ $K'_6{}^{(8)}$ $K'_7{}^{(8)}$;

即

$-K_0^{(8)}$ $(K_1^{(8)})^{-1}$ $(K_2^{(8)})^{-1}$ $-K_3^{(8)}$ $(K_4^{(8)})^{-1}$ $-K_5^{(8)}$ $-K_6^{(8)}$ $(K_7^{(8)})^{-1}$

$-K_0^{(7)}$ $(K_2^{(7)})^{-1}$ $(K_1^{(7)})^{-1}$ $-K_3^{(7)}$ $(K_4^{(7)})^{-1}$ $-K_6^{(7)}$ $-K_5^{(7)}$ $(K_7^{(7)})^{-1}$

$-K_0^{(6)}$ $(K_2^{(6)})^{-1}$ $(K_1^{(6)})^{-1}$ $-K_3^{(6)}$ $(K_4^{(6)})^{-1}$ $-K_6^{(6)}$ $-K_5^{(6)}$ $(K_7^{(6)})^{-1}$

$-K_0^{(5)}$ $(K_2^{(5)})^{-1}$ $(K_1^{(5)})^{-1}$ $-K_3^{(5)}$ $(K_4^{(5)})^{-1}$ $-K_6^{(5)}$ $-K_5^{(5)}$ $(K_7^{(5)})^{-1}$

$-K_0^{(4)}$ $(K_2^{(4)})^{-1}$ $(K_1^{(4)})^{-1}$ $-K_3^{(4)}$ $(K_4^{(4)})^{-1}$ $-K_6^{(4)}$ $-K_5^{(4)}$ $(K_7^{(4)})^{-1}$

$-K_0^{(3)}$ $(K_2^{(3)})^{-1}$ $(K_1^{(3)})^{-1}$ $-K_3^{(3)}$ $(K_4^{(3)})^{-1}$ $-K_6^{(3)}$ $-K_5^{(3)}$ $(K_7^{(3)})^{-1}$

$-K_0^{(2)}$ $(K_2^{(2)})^{-1}$ $(K_1^{(2)})^{-1}$ $-K_3^{(2)}$ $(K_4^{(2)})^{-1}$ $-K_6^{(2)}$ $-K_5^{(2)}$ $(K_7^{(2)})^{-1}$

$-K_0^{(1)}$ $(K_2^{(1)})^{-1}$ $(K_1^{(1)})^{-1}$ $-K_3^{(1)}$ $(K_4^{(1)})^{-1}$ $-K_6^{(1)}$ $-K_5^{(1)}$ $(K_7^{(1)})^{-1}$

$-K_0^{(0)}$ $(K_1^{(0)})^{-1}$ $(K_2^{(0)})^{-1}$ $-K_3^{(0)}$ $(K_4^{(0)})^{-1}$ $-K_5^{(0)}$ $-K_6^{(0)}$ $(K_7^{(0)})^{-1}$

解密密钥由发送方采用安全途径传输给接收方;

●加密部分采用了下列步骤:

输入:一个128位的明文分组X,

72个16位的加密子密钥$K_i^{(j)}$ (0≤i≤7,0≤j≤8);

(1)把X划分为8个16位的子分组$X_0,X_1,\ldots,X_7$,

令$X_0^{(0)}, X_1^{(0)}, \ldots, X_7^{(0)} \leftarrow X_0, X_1, \ldots, X_7$;

(2)置j←0;

(3)计算

$A\leftarrow X_0^{(j)}[+]K_0^{(j)}$,$B\leftarrow X_1^{(j)}\odot K_1^{(j)}$,$C\leftarrow X_2^{(j)}\odot K_2^{(j)}$,$D\leftarrow X_3^{(j)}[+]K_3^{(j)}$,

$E\leftarrow X_4^{(j)}\odot K_4^{(j)}$,$F\leftarrow X_5^{(j)}[+]K_5^{(j)}$,$G\leftarrow X_6^{(j)}[+]K_6^{(j)}$,$H\leftarrow X_7^{(j)}\odot K_7^{(j)}$,

$I\leftarrow A\oplus C$,　$J\leftarrow B\oplus D$,　$Z\leftarrow E\oplus G$,　$L\leftarrow F\oplus H$,

$M\leftarrow I[+]J$,$N\leftarrow Z\odot M$,$Q\leftarrow L[+]N$,$P\leftarrow M\odot Q$,

$R\leftarrow I\odot Q$,$S\leftarrow L[+]P$,

$X_0^{(j+1)}\leftarrow A\oplus P$,　$X_1^{(j+1)}\leftarrow C\oplus P$,　$X_4^{(j+1)}\leftarrow E\oplus S$,　$X_5^{(j+1)}\leftarrow G\oplus S$,

$X_2^{(j+1)}\leftarrow B\oplus R$,　$X_3^{(j+1)}\leftarrow D\oplus R$,　$X_6^{(j+1)}\leftarrow F\oplus Q$,　$X_7^{(j+1)}\leftarrow H\oplus Q$;

(4) 令j←j+1;

(5) 如果j≤7,则转至(3),否则,下一步;

(6) 令$T\leftarrow X_1^{(8)}$,$X_1^{(8)}\leftarrow X_2^{(8)}$,$X_2^{(8)}\leftarrow T$,

令$T\leftarrow X_5^{(8)}$,$X_5^{(8)}\leftarrow X_6^{(8)}$,$X_6^{(8)}\leftarrow T$;

(7) 计算

$Y_0\leftarrow X_0^{(8)}[+]K_0^{(8)}$,$Y_1\leftarrow X_1^{(8)}\odot K_1^{(8)}$,$Y_2\leftarrow X_2^{(8)}\odot K_2^{(8)}$,$Y_3\leftarrow X_3^{(8)}[+]K_3^{(8)}$,

$Y_4\leftarrow X_4^{(8)}\odot K_4^{(8)}$,$Y_5\leftarrow X_5^{(8)}[+]K_5^{(8)}$,$Y_6\leftarrow X_6^{(8)}[+]K_6^{(8)}$,$Y_7\leftarrow X_7^{(8)}\odot K_7^{(8)}$;

(8) 连接$Y_0,Y_1,\ldots,Y_7$为Y;

输出:128位的密文分组Y;

●解密部分采用了下列步骤:

输入:一个128位的密文分组Y,

72个16位的解密子密钥$K'^{(j)}_i$(0≤i≤7,0≤j≤8);

(1) 把Y划分为8个16位的子分组$Y_0,Y_1,\ldots,Y_7$,

令$X'^{(0)}_0,X'^{(0)}_1,\ldots,X'^{(0)}_7\leftarrow Y_0,Y_1,\ldots,Y_7$;

(2) 置j←0;

(3) 计算

$A\leftarrow X'^{(j)}_0[+]K'^{(j)}_0$,　$B\leftarrow X'^{(j)}_1\odot K'^{(j)}_1$,　$C\leftarrow X'^{(j)}_2\odot K'^{(j)}_2$,　$D\leftarrow X'^{(j)}_3[+]K'^{(j)}_3$,

$E\leftarrow X'^{(j)}_4\odot K'^{(j)}_4$,　$F\leftarrow X'^{(j)}_5[+]K'^{(j)}_5$,　$G\leftarrow X'^{(j)}_6[+]K'^{(j)}_6$,　$H\leftarrow X'^{(j)}_7\odot K'^{(j)}_7$,

$I\leftarrow A\oplus C$,　$J\leftarrow B\oplus D$,　$Z\leftarrow E\oplus G$,　$L\leftarrow F\oplus H$,

$M\leftarrow I[+]J$,$N\leftarrow Z\odot M$,$Q\leftarrow L[+]N$,$P\leftarrow M\odot Q$,

$R\leftarrow I\odot Q$,$S\leftarrow L[+]P$,

$X'^{(j+1)}_0\leftarrow A\oplus P$,　$X'^{(j+1)}_1\leftarrow C\oplus P$,　$X'^{(j+1)}_4\leftarrow E\oplus S$,　$X'^{(j+1)}_5\leftarrow G\oplus S$,

$X'^{(j+1)}_2\leftarrow B\oplus R$,　$X'^{(j+1)}_3\leftarrow D\oplus R$,　$X'^{(j+1)}_6\leftarrow F\oplus Q$,　$X'^{(j+1)}_7\leftarrow H\oplus Q$;

(4) 令j←j+1;

(5) 如果j≤7,则转至(3),否则,下一步;

(6) 令$T\leftarrow X'^{(8)}_1$,　$X'^{(8)}_1\leftarrow X'^{(8)}_2$,　$X'^{(8)}_2\leftarrow T$,

令$T\leftarrow X'^{(8)}_5$,　$X'^{(8)}_5\leftarrow X'^{(8)}_6$,　$X'^{(8)}_6\leftarrow T$;

(7) 计算

$Y'_0\leftarrow X'^{(8)}_0[+]K'^{(8)}_0$,　$Y'_1\leftarrow X'^{(8)}_1\odot K'^{(8)}_1$,　$Y'_2\leftarrow X'^{(8)}_2\odot K'^{(8)}_2$,　$Y'_3\leftarrow X'^{(8)}_3[+]K'^{(8)}_3$,

$Y'_4\leftarrow X'^{(8)}_4\odot K'^{(8)}_4$,　$Y'_5\leftarrow X'^{(8)}_5[+]K'^{(8)}_5$,　$Y'_6\leftarrow X'^{(8)}_6[+]K'^{(8)}_6$,　$Y'_7\leftarrow X'^{(8)}_7\odot K'^{(8)}_7$;

(8)连接$Y'_0,Y'_1,...,Y'_7$为Y′,令X←Y′;

输出:128位的明文分组X。

基于三个群运算的128位对称加密方法

(一)技术领域

[0001] 对称加密方法属于密码技术和计算机技术领域,是电子金融安全、电子商务安全、电子政务安全、可信计算和网络信息安全的核心技术之一。

(二)背景技术

[0002] 密码技术是一门古来而又活跃的技术,其发展大致可分为古典密码技术、对称密码技术和非对称密码技术三个阶段。目前是对称密码和非对称密码共存的局面,且非对称密码主要用来加密对称密码的密钥。对称密码又可进一步分为分组密码和流密码。下文中,如果没有特别说明,则对称密码均指分组密码。

[0003] 1977年,美国国家标准技术局(National Institute of Standards and Technology,简称NIST)提出DES数据加密标准——一个对称加密算法(FIPS 46-3,NIST,1977),其分组长度为64比特,密钥长度为56比特。1990年,来学嘉等学者提出了IDEA对称加密算法(X.Lai and J.Massey,A Proposal for a New BlocK Encryption Standard,Proc.of Advances in Cryptology-EUROCRYPT'90,Berlin:Springer-Verlag,1991),其分组长度为64比特,密钥长度为128比特。2001年,NIST颁布了DES的替代版本——AES高级加密标准(FIPS 197,NIST,2001),其分组长度为128比特,密钥长度为128、192或256比特。

[0004] 2006年,我国颁布SMS4对称密码算法(国家密码管理局公告,第7号,2006年1月。2012年改称为SM4),其分组长度为128比特,密钥长度为128比特,主要用于无线网。2012年,我国颁布SM1序列密码算法(国家密码管理局公告,第23号,2012年3月),它是一个流密码,其初始密钥长度为128比特,初始向量长度也为128比特。

[0005] 上述5个对称密码体制中,DES和IDEA已不再使用,AES在国际上使用最为普遍,SM1和SM4主要在国内使用,且有特定的应用环境要求。

(三)发明内容

[0006] 本发明用于计算机和通信网络中字符、文字、音频、图像、视频等各种数据与文件的加密和解密,以确保数据、文件内容的保密存储与传输,可广泛应用于电子金融、电子商务、电子政务和互联网中。

[0007] 密码技术是保障网络安全的核心技术所在,是维护网络主权的关键抓手所在。本发明希望我们国家在对称密码领域能够拥有更多的自主原创技术,以供不同环境、不同条件下的用户选择和应用。

[0008] 在本文中,符号←表示把右边的常数值或表达式值赋给左边的变量;"位"表示一个二进制位(也称为比特,其值为0或1);%代表模运算;⊕代表两个16位运算数的模2加法(即比特异或);[+]代表两个16位运算数的模2^{16}加法;○代表两个16位运算数的模$2^{16}+1$乘法;$K_i^{(j)}$和$K'_i{}^{(j)}$分别代表加密和解密第j轮第i个子密钥;$-K_i^{(j)}$表示$K_i^{(j)}$模2^{16}的加法逆元;$(K_i^{(j)})^{-1}$表示$K_i^{(j)}$模$2^{16}+1$的乘法逆元。

[0009] 特别,对于模$2^{16}+1$乘法,元素2^{16}(即65536)的逆元仍然是2^{16},又由于2^{16}的低16位全

部是零，因此，我们使用16位全零元素代表2^{16}以及2^{16}的乘法逆元，并且，任何一个元素和16位全零元素做模$2^{16}+1$乘法运算时，16位全零元素应被视作2^{16}，任何一个元素和16位全零元素做模2^{16}加法运算时，16位全零元素仍被视作0。

[0010] 3.1基于对称加密的保密通信

[0011] 在对称密码体制中，加密部件与解密部件一般是相同的，加密初始密钥和解密初始密钥也是相同的。在本发明中，亦是这样。

[0012] 在本文中，把加密之前的文件或数据叫作明文或明文分组，加密之后的文件或数据叫作密文或密文分组。

[0013] 假设用户U(发送方)欲通过公共网络向用户V(接收方)传输一个文件或数据，且以保密的方式进行。那么，其过程如下：

[0014] ①密钥生成：用户U随机产生一个256位的初始密钥K，从K得到加密密钥$K_i^{(j)}$，进而，从$K_i^{(j)}$得到解密密钥$K'_i^{(j)}$，并把解密密钥$K'_i^{(j)}$通过安全途径传送给用户V，其中，i(0≤i≤7)代表子密钥的序号，j(0≤j≤8)代表迭代的序号。

[0015] ②加密操作：用户U输入加密密钥$K_i^{(j)}$和128位的明文分组X到加密部件中，得到128位的密文分组Y，并通过公共网络把Y传送给用户V。

[0016] ③解密操作：用户V接收到U发来的128位密文分组Y后，输入解密密钥$K'_i^{(j)}$和密文分组Y到解密部件中，恢复出明文分组X。

[0017] 3.2本发明的技术方案

[0018] 本技术方案，由密钥生成、加密和解密等三个部分组成。

[0019] 根据本发明，可制造密钥生成芯片、加密芯片和解密芯片，或者开发密钥生成软件、加密软件和解密软件等。因此，本发明是一种生产对称密码产品所必须遵循的基本原理与技术方案，而不是物理产品本身。

[0020] 3.2.1密钥生成部分

[0021] 每个16位子密钥$K_i^{(j)}$或$K'_i^{(j)}$由初始密钥K变换而来。密钥生成部分的实现方法如下：

[0022] 输入：一个256位的随机初始密钥K；

[0023] (1)置j←0；

[0024] (2)把K最左边128位划分为8个16位的子密钥，分别赋给$K_0^{(j)}$，$K_1^{(j)}$，...，$K_7^{(j)}$；

[0025] (3)把K循环左移25位；

[0026] (4)令j←j+1；

[0027] (5)如果j≤8，则转至(2)，否则，下一步；

[0028] (6)令$K'_0{}^{(0)}$，$K'_0{}^{(1)}$，...，$K'_0{}^{(8)}\leftarrow -K_0^{(8)}$，$-K_0^{(7)}$，...，$-K_0^{(0)}$，

[0029] 令$K'_3{}^{(0)}$，$K'_3{}^{(1)}$，...，$K'_3{}^{(8)}\leftarrow -K_3^{(8)}$，$-K_3^{(7)}$，...，$-K_3^{(0)}$，

[0030] 令$K'_4{}^{(0)}$，$K'_4{}^{(1)}$，...，$K'_4{}^{(8)}\leftarrow (K_4^{(8)})^{-1}$，$(K_4^{(7)})^{-1}$，...，$(K_4^{(0)})^{-1}$，

[0031] 令$K'_7{}^{(0)}$，$K'_7{}^{(1)}$，...，$K'_7{}^{(8)}\leftarrow (K_7^{(8)})^{-1}$，$(K_7^{(7)})^{-1}$，...，$(K_7^{(0)})^{-1}$；

[0032] (7)令$K'_1{}^{(0)}\leftarrow (K_1^{(8)})^{-1}$，$K'_1{}^{(8)}\leftarrow (K_1^{(0)})^{-1}$，$K'_2{}^{(0)}\leftarrow (K_2^{(8)})^{-1}$，$K'_2{}^{(8)}\leftarrow (K_2^{(0)})^{-1}$，

[0033] 令$K'_5{}^{(0)}\leftarrow -K_5^{(8)}$，$K'_5{}^{(8)}\leftarrow -K_5^{(0)}$，$K'_6{}^{(0)}\leftarrow -K_6^{(8)}$，$K'_6{}^{(8)}\leftarrow -K_6^{(0)}$；

[0034] (8)令$K'_1{}^{(1)}$，$K'_1{}^{(2)}$，...，$K'_1{}^{(7)}\leftarrow (K_2^{(7)})^{-1}$，$(K_2^{(6)})^{-1}$，...，$(K_2^{(1)})^{-1}$，

[0035] 令$K'_2{}^{(1)}$，$K'_2{}^{(2)}$，...，$K'_2{}^{(7)}\leftarrow (K_1^{(7)})^{-1}$，$(K_1^{(6)})^{-1}$，...，$(K_1^{(1)})^{-1}$，

[0036] 令$K'_5^{(1)},K'_5^{(2)},\ldots,K'_5^{(7)} \leftarrow -K_6^{(7)},-K_6^{(6)},\ldots,-K_6^{(1)}$,

[0037] 令$K'_6^{(1)},K'_6^{(2)},\ldots,K'_6^{(7)} \leftarrow -K_5^{(7)},-K_5^{(6)},\ldots,-K_5^{(1)}$;

[0038] 输出:加密密钥(由72个子密钥组成)

$$\begin{array}{cccccccc} K_0^{(0)} & K_1^{(0)} & K_2^{(0)} & K_3^{(0)} & K_4^{(0)} & K_5^{(0)} & K_6^{(0)} & K_7^{(0)} \\ K_0^{(1)} & K_1^{(1)} & K_2^{(1)} & K_3^{(1)} & K_4^{(1)} & K_5^{(1)} & K_6^{(1)} & K_7^{(1)} \\ K_0^{(2)} & K_1^{(2)} & K_2^{(2)} & K_3^{(2)} & K_4^{(2)} & K_5^{(2)} & K_6^{(2)} & K_7^{(2)} \\ K_0^{(3)} & K_1^{(3)} & K_2^{(3)} & K_3^{(3)} & K_4^{(3)} & K_5^{(3)} & K_6^{(3)} & K_7^{(3)} \end{array}$$

[0039]

$$\begin{array}{cccccccc} K_0^{(4)} & K_1^{(4)} & K_2^{(4)} & K_3^{(4)} & K_4^{(4)} & K_5^{(4)} & K_6^{(4)} & K_7^{(4)} \\ K_0^{(5)} & K_1^{(5)} & K_2^{(5)} & K_3^{(5)} & K_4^{(5)} & K_5^{(5)} & K_6^{(5)} & K_7^{(5)} \\ K_0^{(6)} & K_1^{(6)} & K_2^{(6)} & K_3^{(6)} & K_4^{(6)} & K_5^{(6)} & K_6^{(6)} & K_7^{(6)} \\ K_0^{(7)} & K_1^{(7)} & K_2^{(7)} & K_3^{(7)} & K_4^{(7)} & K_5^{(7)} & K_6^{(7)} & K_7^{(7)} \\ K_0^{(8)} & K_1^{(8)} & K_2^{(8)} & K_3^{(8)} & K_4^{(8)} & K_5^{(8)} & K_6^{(8)} & K_7^{(8)}; \end{array}$$

[0040] 解密密钥(由72个子密钥组成)

$$\begin{array}{cccccccc} K'_0^{(0)} & K'_1^{(0)} & K'_2^{(0)} & K'_3^{(0)} & K'_4^{(0)} & K'_5^{(0)} & K'_6^{(0)} & K'_7^{(0)} \\ K'_0^{(1)} & K'_1^{(1)} & K'_2^{(1)} & K'_3^{(1)} & K'_4^{(1)} & K'_5^{(1)} & K'_6^{(1)} & K'_7^{(1)} \\ K'_0^{(2)} & K'_1^{(2)} & K'_2^{(2)} & K'_3^{(2)} & K'_4^{(2)} & K'_5^{(2)} & K'_6^{(2)} & K'_7^{(2)} \\ K'_0^{(3)} & K'_1^{(3)} & K'_2^{(3)} & K'_3^{(3)} & K'_4^{(3)} & K'_5^{(3)} & K'_6^{(3)} & K'_7^{(3)} \\ K'_0^{(4)} & K'_1^{(4)} & K'_2^{(4)} & K'_3^{(4)} & K'_4^{(4)} & K'_5^{(4)} & K'_6^{(4)} & K'_7^{(4)} \\ K'_0^{(5)} & K'_1^{(5)} & K'_2^{(5)} & K'_3^{(5)} & K'_4^{(5)} & K'_5^{(5)} & K'_6^{(5)} & K'_7^{(5)} \end{array}$$

[0041]

[0042]

$$\begin{array}{cccccccc} K'_0^{(6)} & K'_1^{(6)} & K'_2^{(6)} & K'_3^{(6)} & K'_4^{(6)} & K'_5^{(6)} & K'_6^{(6)} & K'_7^{(6)} \\ K'_0^{(7)} & K'_1^{(7)} & K'_2^{(7)} & K'_3^{(7)} & K'_4^{(7)} & K'_5^{(7)} & K'_6^{(7)} & K'_7^{(7)} \\ K'_0^{(8)} & K'_1^{(8)} & K'_2^{(8)} & K'_3^{(8)} & K'_4^{(8)} & K'_5^{(8)} & K'_6^{(8)} & K'_7^{(8)}; \end{array}$$

[0043] 即

$$\begin{array}{cccccccc} -K_0^{(8)} & (K_1^{(8)})^{-1} & (K_2^{(8)})^{-1} & -K_3^{(8)} & (K_4^{(8)})^{-1} & -K_5^{(8)} & -K_6^{(8)} & (K_7^{(8)})^{-1} \\ -K_0^{(7)} & (K_2^{(7)})^{-1} & (K_1^{(7)})^{-1} & -K_3^{(7)} & (K_4^{(7)})^{-1} & -K_6^{(7)} & -K_5^{(7)} & (K_7^{(7)})^{-1} \\ -K_0^{(6)} & (K_2^{(6)})^{-1} & (K_1^{(6)})^{-1} & -K_3^{(6)} & (K_4^{(6)})^{-1} & -K_6^{(6)} & -K_5^{(6)} & (K_7^{(6)})^{-1} \\ -K_0^{(5)} & (K_2^{(5)})^{-1} & (K_1^{(5)})^{-1} & -K_3^{(5)} & (K_4^{(5)})^{-1} & -K_6^{(5)} & -K_5^{(5)} & (K_7^{(5)})^{-1} \end{array}$$

[0044]

$$\begin{array}{cccccccc} -K_0^{(4)} & (K_2^{(4)})^{-1} & (K_1^{(4)})^{-1} & -K_3^{(4)} & (K_4^{(4)})^{-1} & -K_6^{(4)} & -K_5^{(4)} & (K_7^{(4)})^{-1} \\ -K_0^{(3)} & (K_2^{(3)})^{-1} & (K_1^{(3)})^{-1} & -K_3^{(3)} & (K_4^{(3)})^{-1} & -K_6^{(3)} & -K_5^{(3)} & (K_7^{(3)})^{-1} \\ -K_0^{(2)} & (K_2^{(2)})^{-1} & (K_1^{(2)})^{-1} & -K_3^{(2)} & (K_4^{(2)})^{-1} & -K_6^{(2)} & -K_5^{(2)} & (K_7^{(2)})^{-1} \\ -K_0^{(1)} & (K_2^{(1)})^{-1} & (K_1^{(1)})^{-1} & -K_3^{(1)} & (K_4^{(1)})^{-1} & -K_6^{(1)} & -K_5^{(1)} & (K_7^{(1)})^{-1} \\ -K_0^{(0)} & (K_1^{(0)})^{-1} & (K_2^{(0)})^{-1} & -K_3^{(0)} & (K_4^{(0)})^{-1} & -K_5^{(0)} & -K_6^{(0)} & (K_7^{(0)})^{-1}。 \end{array}$$

[0045] 解密密钥由发送方(用户U)采用安全途径传输给接收方(用户V)。

[0046] 3.2.2加密部分

[0047] 加密部分供发送方使用，用来对明文分组进行加密。其实现方法如下：

[0048] 输入：一个128位的明文分组X，

[0049] 72个16位的加密子密钥$K_i^{(j)}$ ($0 \leqslant i \leqslant 7, 0 \leqslant j \leqslant 8$)；

[0050] (1)把X划分为8个16位的子分组$X_0, X_1, \ldots, X_7$，

[0051] 令$X_0^{(0)}, X_1^{(0)}, \ldots, X_7^{(0)} \leftarrow X_0, X_1, \ldots, X_7$；

[0052] (2)置$j \leftarrow 0$；

[0053] (3)计算

$$A \leftarrow X_0^{(j)} [+] K_0^{(j)}, \quad B \leftarrow X_1^{(j)} \odot K_1^{(j)}, \quad C \leftarrow X_2^{(j)} \odot K_2^{(j)}, \quad D \leftarrow X_3^{(j)} [+] K_3^{(j)},$$

$$E \leftarrow X_4^{(j)} \odot K_4^{(j)}, \quad F \leftarrow X_5^{(j)} [+] K_5^{(j)}, \quad G \leftarrow X_6^{(j)} [+] K_6^{(j)}, \quad H \leftarrow X_7^{(j)} \odot K_7^{(j)},$$

$$I \leftarrow A \oplus C, \quad J \leftarrow B \oplus D, \quad Z \leftarrow E \oplus G, \quad L \leftarrow F \oplus H,$$

[0054]

$$M \leftarrow I [+] J, \quad N \leftarrow Z \odot M, \quad Q \leftarrow L [+] N, \quad P \leftarrow M \odot Q,$$

$$R \leftarrow I \odot Q, \quad S \leftarrow L [+] P,$$

$$X_0^{(j+1)} \leftarrow A \oplus P, \quad X_1^{(j+1)} \leftarrow C \oplus P, \quad X_4^{(j+1)} \leftarrow E \oplus S, \quad X_5^{(j+1)} \leftarrow G \oplus S,$$

$$X_2^{(j+1)} \leftarrow B \oplus R, \quad X_3^{(j+1)} \leftarrow D \oplus R, \quad X_6^{(j+1)} \leftarrow F \oplus Q, \quad X_7^{(j+1)} \leftarrow H \oplus Q;$$

[0055] (4)令$j \leftarrow j+1$；

[0056] (5)如果$j \leqslant 7$，则转至(3)，否则，下一步；

[0057] (6)令$T \leftarrow X_1^{(8)}, X_1^{(8)} \leftarrow X_2^{(8)}, X_2^{(8)} \leftarrow T$，

[0058] 令$T \leftarrow X_5^{(8)}, X_5^{(8)} \leftarrow X_6^{(8)}, X_6^{(8)} \leftarrow T$；

[0059] (7)计算

[0060] $Y_0 \leftarrow X_0^{(8)} [+] K_0^{(8)}, Y_1 \leftarrow X_1^{(8)} \odot K_1^{(8)}, Y_2 \leftarrow X_2^{(8)} \odot K_2^{(8)}, Y_3 \leftarrow X_3^{(8)} [+] K_3^{(8)}$，

[0061] $Y_4 \leftarrow X_4^{(8)} \odot K_4^{(8)}, Y_5 \leftarrow X_5^{(8)} [+] K_5^{(8)}, Y_6 \leftarrow X_6^{(8)} [+] K_6^{(8)}, Y_7 \leftarrow X_7^{(8)} \odot K_7^{(8)}$；

[0062] (8)连接$Y_0, Y_1, \ldots, Y_7$为Y；

[0063] 输出：128位的密文分组Y。

[0064] 3.2.3解密部分

[0065] 解密部分供接收方使用，用来对密文分组进行解密。其实现方法如下：

[0066] 输入：一个128位的密文分组Y，

[0067] 72个16位的解密子密钥$K'_i^{(j)}$ ($0 \leqslant i \leqslant 7, 0 \leqslant j \leqslant 8$)；

[0068] (1)把Y划分为8个16位的子分组$Y_0, Y_1, \ldots, Y_7$，

[0069] 令$X'_0{}^{(0)}, X'_1{}^{(0)}, \ldots, X'_7{}^{(0)} \leftarrow Y_0, Y_1, \ldots, Y_7$；

[0070] (2)置$j \leftarrow 0$；

[0071] (3)计算

$A \leftarrow X'_0^{(j)} [+] K'_0^{(j)}$, $B \leftarrow X'_1^{(j)} \odot K'_1^{(j)}$, $C \leftarrow X'_2^{(j)} \odot K'_2^{(j)}$, $D \leftarrow X'_3^{(j)} [+] K'_3^{(j)}$,

$E \leftarrow X'_4^{(j)} \odot K'_4^{(j)}$, $F \leftarrow X'_5^{(j)} [+] K'_5^{(j)}$, $G \leftarrow X'_6^{(j)} [+] K'_6^{(j)}$, $H \leftarrow X'_7^{(j)} \odot K'_7^{(j)}$,

$I \leftarrow A \oplus C$, $J \leftarrow B \oplus D$, $Z \leftarrow E \oplus G$, $L \leftarrow F \oplus H$,

[0072] $M \leftarrow I [+] J$, $N \leftarrow Z \odot M$, $Q \leftarrow L [+] N$, $P \leftarrow M \odot Q$,

$R \leftarrow I \odot Q$, $S \leftarrow L [+] P$,

$X'_0^{(j+1)} \leftarrow A \oplus P$, $X'_1^{(j+1)} \leftarrow C \oplus P$, $X'_4^{(j+1)} \leftarrow E \oplus S$, $X'_5^{(j+1)} \leftarrow G \oplus S$,

$X'_2^{(j+1)} \leftarrow B \oplus R$, $X'_3^{(j+1)} \leftarrow D \oplus R$, $X'_6^{(j+1)} \leftarrow F \oplus Q$, $X'_7^{(j+1)} \leftarrow H \oplus Q$;

[0073] (4) 令j←j+1;

[0074] (5) 如果j≤7,则转至(3),否则,下一步;

[0075] (6) 令$T \leftarrow X'_1^{(8)}$,$X'_1^{(8)} \leftarrow X'_2^{(8)}$,$X'_2^{(8)} \leftarrow T$,

[0076] 令$T \leftarrow X'_5^{(8)}$,$X'_5^{(8)} \leftarrow X'_6^{(8)}$,$X'_6^{(8)} \leftarrow T$;

[0077] (7) 计算

[0078] $Y'_0 \leftarrow X'_0^{(8)} [+] K'_0^{(8)}$,$Y'_1 \leftarrow X'_1^{(8)} \odot K'_1^{(8)}$,$Y'_2 \leftarrow X'_2^{(8)} \odot K'_2^{(8)}$,$Y'_3 \leftarrow X'_3^{(8)} [+] K'_3^{(8)}$,

[0079] $Y'_4 \leftarrow X'_4^{(8)} \odot K'_4^{(8)}$,$Y'_5 \leftarrow X'_5^{(8)} [+] K'_5^{(8)}$,$Y'_6 \leftarrow X'_6^{(8)} [+] K'_6^{(8)}$,$Y'_7 \leftarrow X'_7^{(8)} \odot K'_7^{(8)}$;

[0080] (8) 连接$Y'_0, Y'_1, \ldots, Y'_7$为Y',令X←Y';

[0081] 输出:128位的明文分组X。

[0082] 3.2.4方法的正确性

[0083] 假设在输出处理之前只有一轮迭代。这并不影响本对称加密方法的正确性,因为我们只需要说明轮函数和输出变换是可逆的即可。

[0084] 首先考虑加密。

[0085] 把明文分组X划分为8个16位的子分组,并赋给$X_0^{(0)}, X_1^{(0)}, \ldots, X_7^{(0)}$。

[0086] 以明文分组$X_i^{(0)}$和加密子密钥$K_i^{(0)}$、$K_i^{(1)}$ (i=0,1,...,7)作为输入,根据3.2.2节,加密轮运算如下:

(00) $A = X_0^{(0)} [+] K_0^{(0)}$ (01) $B = X_1^{(0)} \odot K_1^{(0)}$

(02) $C = X_2^{(0)} \odot K_2^{(0)}$ (03) $D = X_3^{(0)} [+] K_3^{(0)}$

(04) $E = X_4^{(0)} \odot K_4^{(0)}$ (05) $F = X_5^{(0)} [+] K_5^{(0)}$

(06) $G = X_6^{(0)} [+] K_6^{(0)}$ (07) $H = X_7^{(0)} \odot K_7^{(0)}$

(08) $I = A \oplus C$ (09) $J = B \oplus D$

(10) $Z = E \oplus G$ (11) $L = F \oplus H$

[0087] (12) $M = I [+] J$ (13) $N = Z \odot M$

(14) $Q = L [+] N$ (15) $P = M \odot Q$

(16) $R = I \odot Q$ (17) $S = L [+] P$

(18) $X_0^{(1)} = A \oplus P$ (19) $X_2^{(1)} = C \oplus P$

(20) $X_4^{(1)} = E \oplus S$ (21) $X_6^{(1)} = G \oplus S$

(22) $X_1^{(1)} = B \oplus R$ (23) $X_3^{(1)} = D \oplus R$

(24) $X_5^{(1)} = F \oplus Q$ (25) $X_7^{(1)} = H \oplus Q$

[0088] 注意，由于只有一轮迭代，因此，$X_1^{(1)}$、$X_2^{(1)}$之间和$X_5^{(1)}$、$X_6^{(1)}$之间无需做交换。

[0089] 进一步，做输出处理：

(00) $Y_0 = X_0^{(1)} [+] K_0^{(1)}$ (01) $Y_1 = X_1^{(1)} \odot K_1^{(1)}$

[0090] (02) $Y_2 = X_2^{(1)} \odot K_2^{(1)}$ (03) $Y_3 = X_3^{(1)} [+] K_3^{(1)}$

(04) $Y_4 = X_4^{(1)} \odot K_4^{(1)}$ (05) $Y_5 = X_5^{(1)} [+] K_5^{(1)}$

(07) $Y_6 = X_6^{(1)} [+] K_6^{(1)}$ (07) $Y_7 = X_7^{(1)} \odot K_7^{(1)}$

[0091] 连接$Y_0, Y_1, \ldots, Y_7$成为密文分组Y。

[0092] 下面考虑解密。

[0093] 把密文分组Y划分为8个16位的子分组，即$Y_0, Y_1, \ldots, Y_7$，并赋给$X'_0^{(0)}, X'_1^{(0)}, \ldots, X'_7^{(0)}$。以密文分组$X'_i^{(0)}$和解密子密钥$Z'_i^{(0)}$、$Z'_i^{(1)}$ (i＝0,1,...,7)作为输入，根据3.2.3节，解密轮运算如下：

[0094] (00) $A' = X'_0^{(0)} [+] (-Z_0^{(1)}) = X_0^{(1)} [+] Z_0^{(1)} [+] (-Z_0^{(1)}) = X_0^{(1)}$

(01) $B' = X'_1^{(0)} \odot (Z_1^{(1)})^{-1} = X_1^{(1)} \odot Z_1^{(1)} \odot (Z_1^{(1)})^{-1} = X_1^{(1)}$

(02) $C'=X'_2^{(0)}\odot(Z_2^{(1)})^{-1}=X_2^{(1)}\odot Z_2^{(1)}\odot(Z_2^{(1)})^{-1}=X_2^{(1)}$

(03) $D'=X'_3^{(0)}[+](-Z_3^{(1)})=X_3^{(1)}[+]Z_3^{(1)}[+](-Z_3^{(1)})=X_3^{(1)}$

(04) $E'=X'_4^{(0)}\odot(Z_4^{(1)})^{-1}=X_4^{(1)}\odot Z_4^{(1)}\odot(Z_4^{(1)})^{-1}=X_4^{(1)}$

(05) $F'=X'_5^{(0)}[+](-Z_5^{(1)})=X_5^{(1)}[+]Z_5^{(1)}[+](-Z_5^{(1)})=X_5^{(1)}$

(06) $G'=X'_6^{(0)}[+](-Z_6^{(1)})=X_6^{(1)}[+]Z_6^{(1)}[+](-Z_6^{(1)})=X_6^{(1)}$

(07) $H'=X'_7^{(0)}\odot(Z_7^{(1)})^{-1}=X_7^{(1)}\odot Z_7^{(1)}\odot(Z_7^{(1)})^{-1}=X_7^{(1)}$

(08) $I'=A'\oplus C'=X_0^{(1)}\oplus X_2^{(1)}=A\oplus C=I$ (09) $J'=B'\oplus D'=X_1^{(1)}\oplus X_3^{(1)}=B\oplus D=J$

[0095] (10) $K'=E'\oplus G'=X_4^{(1)}\oplus X_6^{(1)}=E\oplus G=K$ (11) $L'=F'\oplus H'=X_5^{(1)}\oplus X_7^{(1)}=F\oplus H=L$

(12) $M'=I'[+]J'=I[+]J=M$ (13) $N'=K'\odot M'=K\odot M=N$

(14) $Q'=L'[+]N'=L[+]N=Q$ (15) $P'=M'\odot Q'=M\odot Q=P$

(16) $R'=I'\odot Q'=I\odot Q=R$ (17) $S'=L'[+]P'=L[+]P=S$

(18) $X'_0^{(1)}=A'\oplus P'=X_0^{(1)}\oplus P=A\oplus P\oplus P=A$ (19) $X'_2^{(1)}=C'\oplus P'=X_2^{(1)}\oplus P=C\oplus P\oplus P=C$

(20) $X'_4^{(1)}=E'\oplus S'=X_4^{(1)}\oplus S=E\oplus S\oplus S=E$ (21) $X'_6^{(1)}=G'\oplus S'=X_6^{(1)}\oplus S=G\oplus S\oplus S=G$

(22) $X'_1^{(1)}=B'\oplus R'=X_1^{(1)}\oplus R=B\oplus R\oplus R=B$ (23) $X'_3^{(1)}=D'\oplus R'=X_3^{(1)}\oplus R=D\oplus R\oplus R=D$

(24) $X'_5^{(1)}=F'\oplus Q'=X_5^{(1)}\oplus Q=F\oplus Q\oplus Q=F$ (25) $X'_7^{(1)}=H'\oplus Q'=X_7^{(1)}\oplus Q=H\oplus Q\oplus Q=H$

[0096] 注意，由于只有一轮迭代，因此，$X'_1{}^{(1)}$、$X'_2{}^{(1)}$之间和$X'_5{}^{(1)}$、$X'_6{}^{(1)}$之间无需做交换。

[0097] 进一步，做输出处理：

[0098] (00) $Y'_0=X'_0{}^{(1)}[+](-Z_0{}^{(0)})=A[+](-Z_0{}^{(0)})=X_0{}^{(0)}[+]Z_0{}^{(0)}[+](-Z_0{}^{(0)})=X_0{}^{(0)}$

[0099] (01) $Y'_1=X'_1{}^{(1)}\odot(Z_1{}^{(0)})^{-1}=B\odot(Z_1{}^{(0)})^{-1}=X_1{}^{(0)}\odot Z_1{}^{(0)}\odot(Z_1{}^{(0)})^{-1}=X_1{}^{(0)}$

[0100] (02) $Y'_2=X'_2{}^{(1)}\odot(Z_2{}^{(0)})^{-1}=C\odot(Z_2{}^{(0)})^{-1}=X_2{}^{(0)}\odot Z_2{}^{(0)}\odot(Z_2{}^{(0)})^{-1}=X_2{}^{(0)}$

[0101] (03) $Y'_3=X'_3{}^{(1)}[+](-Z_3{}^{(0)})=D[+](-Z_3{}^{(0)})=X_3{}^{(0)}[+]Z_3{}^{(0)}[+](-Z_3{}^{(0)})=X_3{}^{(0)}$

[0102] (04) $Y'_4=X'_4{}^{(1)}\odot(Z_4{}^{(0)})^{-1}=E\odot(Z_4{}^{(0)})^{-1}=X_4{}^{(0)}\odot Z_4{}^{(0)}\odot(Z_4{}^{(0)})^{-1}=X_4{}^{(0)}$

[0103] (05) $Y'_5=X'_5{}^{(1)}[+](-Z_5{}^{(0)})=F[+](-Z_5{}^{(0)})=X_5{}^{(0)}[+]Z_5{}^{(0)}[+](-Z_5{}^{(0)})=X_5{}^{(0)}$

[0104] (06) $Y'_6=X'_6{}^{(1)}[+](-Z_6{}^{(0)})=G[+](-Z_6{}^{(0)})=X_6{}^{(0)}[+]Z_6{}^{(0)}[+](-Z_6{}^{(0)})=X_6{}^{(0)}$

[0105] (07) $Y'_7=X'_7{}^{(1)}\odot(Z_7{}^{(0)})^{-1}=H\odot(Z_7{}^{(0)})^{-1}=X_7{}^{(0)}\odot Z_7{}^{(0)}\odot(Z_7{}^{(0)})^{-1}=X_7{}^{(0)}$

[0106] 连接$Y'_0,Y'_1,\ldots,Y'_7$即$X_0{}^{(0)},X_1{}^{(0)},\ldots,X_7{}^{(0)}$成为明文分组X。

[0107] 因此，本对称加密方法是正确的。

[0108] 3.3优点和积极效果

[0109] 3.3.1安全性高

[0110] 本技术方案采用的明文分组长度为128比特，与AES相同，另外，还继承了IDEA抗差分分析的特征，因此，安全性高。

[0111] 3.3.2运算速度快

[0112] 本技术方案用来加密128比特的明文分组只需26个群运算，比IDEA方法少了2个，因此，运算速度较快。

[0113] 3.3.3技术可以公开

[0114] 本技术方案完全可以公开，用户只需保证初始密钥不外泄，就可以完全保证密文的安全。

[0115] 3.3.6适用性强

[0116] 本技术方案既可用于无线网中数据的加密，也可用于有线网中数据的加密。

(四)具体实施方式

[0117] 基于三个群运算的128位对称加密方法的特点是加密密钥和解密密钥由同一个初始密钥得到，初始密钥长度为256比特，分组长度为128比特，相对于非对称密码方法来讲，加密速度和解密速度都明显快很多。

[0118] 发送方随机产生初始密钥，进而推导出加密密钥和解密密钥，并把解密密钥通过秘密途径传输给接收方。

[0119] 本加密方法可以用逻辑电路芯片或程序语言来实现，它包括三部分：①根据3.2.1节开发出密钥生成芯片或密钥生成软件模块，供发送方(即用户U)使用；②根据3.2.2节开发出加密芯片或加密软件模块，供发送方使用；③根据3.2.3节开发出解密芯片或解密软件模块，供接收方(即用户V)使用。