دانشکده مهندسی

گروه برق

پایان نامه کارشناسی

گرایش : الکترونیک

عنوان :

پیاده سازی و بهینه سازی الگوریتم رمزنگاری DES جهت شناسایی کلید رمز

استاد راهنما : دکتر وحید رشتچی

نگارش : شهرزاد حدایق پرست

مهر ۹۲

فهرست

فهرست تصاویر

خلاصه

پیاده سازی و بهینه سازی الگوریتم رمز نگاری DES[1] جهت شناسایی کلید رمز

شهرزاد حدایق پرست

الگوریتم رمزنگاری DES، سابقا به عنوان الگوریتم غالب برای کد کردن اطلاعات الکترونیکی مورد استفاده قرار می گرفت. بنا به درخواست NBS [2] مبنی بر پیشنهاد الگوریتمی مناسب جهت حفاظت از اطلاعات الکترونیکی حساس و طبقه بندی نشده دولتی[3]، شرکت IBM [4] در اوایل دهه ۱۹۷۰ این الگوریتم را به NBS ارائه داد. در سال ۱۹۷۶، NBS سرانجام پس از بررسی و نتیجه گیری با NSA [5]، نسخه اصلاح شده این الگوریتم را انتخاب کرد و این نسخه برای ایالات متحده آمریکا به عنوان FIPS [6]، در سال ۱۹۷۷ منتشر گردید.

الگوریتم DES می تواند در نرم افزار یا در سخت افزار خالص پیاده سازی گردد. جهت پیاده سازی با سخت-افزار FPGA [7] ها راه حل سریعتری را ارائه می دهند. در این پروژه، الگوریتم DES با FPGA و زبان برنامه نویسی VHDL[8] اجرا شده است. برای نوشتن و بهینه سازی کد VHDL از نرم افزار QuartusII11.1 و برای شبیه سازی از نرم افزار modelsim 10.0 استفاده شده است.

در این پروژه پنج کد VHDL برای الگوریتم DES نوشته شده اند که هدف آن ها کاهش $\underline{logic - element \times \text{زمان}}$ ، جهت شناسایی سریعتر کلید رمز می باشد.

[1] Data Encryption Standard
[2] National Bureau of Standards
[3] Unclassified
[4] International Business Machines
[5] National Security Agency
[6] Federal Information Processing Standard
[7] Field Programmable Gate Arrays
[8] Very High Speed Integrated Circuit Hardware Description language

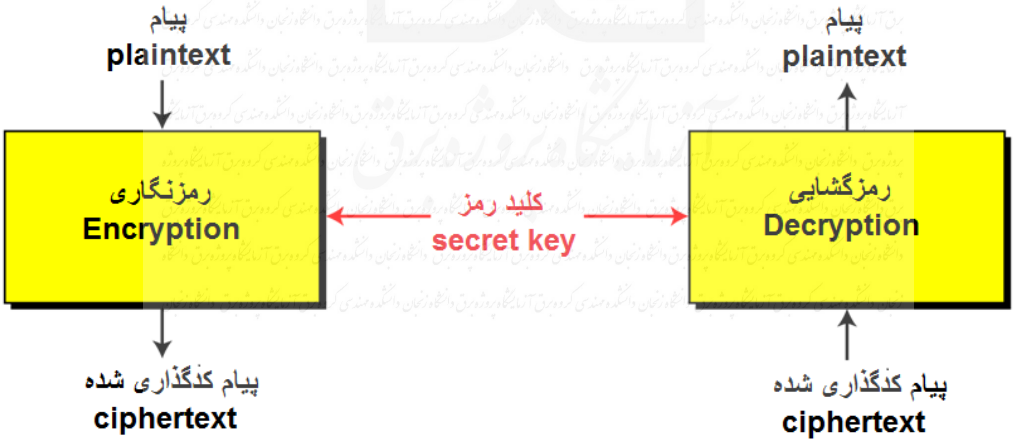
فصل ۱

مقدمه

امروزه دو نوع اصلی رمزنگاری وجود دارد :

- رمزنگاری کلید متقارن[۱]
- رمزنگاری کلید نامتقارن[۲]

رمزنگاری کلید متقارن، نوع قدیمی تر رمزنگاری است در حالی که رمزنگاری کلید نامتقارن پس از دهه ۱۹۷۰ میلادی به طور عمده مورد استفاده قرار گرفت. سابقه رمزنگاری کلید متقارن که در این پروژه نیز مورد نظر ما است، به دوران مصر باستان یا حتی قدیمی تر از آن بازمی گردد که برای رمزنگاری[۳] و رمزگشایی[۴] ، تنها از یک کلید استفاده می کند و به همین دلیل واژه کلید متقارن برای آن انتخاب شده است. از جنبه امنیتی مخفی ماندن کلید رمز ضروری است.



شکل ۱ : رمزنگاری کلید متقارن

---

[۱] symmetric key یا secret key
[۲] asymmetric key یا public key
[۳] encryption
[۴] decryption

برای رمزنگاری کلید متقارن دو تکنیک اصلی مورد استفاده قرار می گیرند :

- جانشینی[۱]
- جایگشت[۲]

جانشینی به طور ساده، نگاشت یک مقدار به مقداری دیگر است در حالی که جایگشت تغییر ترتیب جایگاه بیت ها است. این دو تکنیک به دفعات در هر تکرار که راند[۳] نامیده می شود، مورد استفاده قرار می گیرند. در حالت کلی، هرچه تعداد راند ها بیشر باشد، امنیت الگوریتم نیز بیشتر است. خاصیت غیرخطی بودن[۴] نیز به رمزنگاری داده شده است تا رمزگشایی را بدون داشتن کلید رمز از لحاظ محاسباتی غیر ممکن سازد. این عمل با استفاده از S-box ها که جداول جانشینی غیرخطی هستند ممکن می شود.

یکی از مشکلات اصلی رمزنگاری کلید متقارن، توزیع کلید رمز است. برای رمزنگاری و رمزگشایی هر دو طرف می بایست کلید رمز را داشته باشند و این کلید باید از طریق کانالی امن مبادله شود، که متاسفانه امر ساده ای نیست.

[۱] substitution
[۲] permutation
[۳] round
[۴] non-linearity

۱-۱ تاریخچه DES

تا چندی پیش، یک رمزنگاری کلید متقارن به نام DES، روش استاندارد برای کد کردن اطلاعات بود. با این وجود، در حال حاضر استانداردی جدید به نام AES [۱] جایگزین آن شده است. DES، یک بلوک رمز به طول ۶۴ بیت[۲] است، به این معنا که در هر زمان قادر به کد کردن ۶۴ بیت از اطلاعات ورودی می باشد.

نتیجه یک پروژه تحقیقاتی در زمینه رمزنگاری توسط شرکت IBM در اواخر دهه ۱۹۶۰موجب بوجود آمدن یک رمزنگاری جدید به نام LUCIFER شد. در اوایل دهه ۱۹۷۰ تصمیم به تجاری کردن LUCIFER گرفته شد و تعدادی تغییرات اساسی در آن بوجود آمد. دراین تغییرات نه تنها IBM بلکه NSA نیز دخالت داشت و نظرات فنی در این زمینه اعمال کرد. نسخه جدید بدست آمده از LUCIFER، به عنوان طرحی برای استاندارد رمزنگاری جدید به NBS ارائه گردید. سرانجام در سال ۱۹۷۷ این طرح با نام Data Encryption Standard - DES (FIPS PUB 46) برگزیده شد.

تعدادی از تغییرات اعمال شده در LUCIFER حتی تا به امروز نیز مورد بحث بوده است. از برجسته ترین آن ها طول کلید[۳] بود. LUCIFER از کلیدی به طول ۱۲۸بیت استفاده می کرد اما برای DES این طول به ۵۶ بیت کاهش یافت. با اینکه کلید ورودی DES دارای ۶۴ بیت است، ۸ بیت باقیمانده به عنوان بیت های تعادل[۴] مورد استفاده قرار می گیرند و هیچ تاثیری از جنبه امنیتی برای DES ندارند. حتی در آن زمان، بسیاری از افراد دلایل قانع کننده ای آوردند مبنی بر اینکه کلید با طول ۵۶بیت براحتی می تواند مورد حمله brute force attack قرار گیرد. ضرورت بررسی بیت های تعادل[۵] نیز همواره مورد سوال بوده است، بدون اینکه جواب قانع کننده ای بدست آید.

موضوع دیگر مورد بحث، جداول S-box بودند، چراکه بطور محرمانه طراحی شدند و هیچ توضیحی در مورد نحوه خاص طراحی آن ها داده نشد. این باعث شد مردم فکر کنند که NSA نوعی دریچه [۶] برای خود ساخته که بتواند بوسیله آن هر گونه اطلاعاتی را بدون داشتن کلید رمز، رمزگشایی کند. همچنین یک بررسی نشان داد که S-box ها در مقابل حمله ای به نام Differential Cryptanalysis، که توسط biham و Shamir در سال ۱۹۹۰ بطور عمومی مطرح شد، مقاوم هستند. این موضوع آشکار

---

[۱] Advanced Encryption Standard
[۲] 64bit block cipher
[۳] key size
[۴] parity bit
[۵] parity checking
[۶] trapdoor

کرد که IBM از چنین حمله ای در سال ۱۹۷۷ اطلاع داشته، ۱۳ سال زودتر! طراحان DES ادعا کردند که دلیل افشا نکردن ویژگی ها و نحوه طراحی DES به خاطر تعدادی از حمله های ممکن بود که تا آن زمان بطور عمومی شناخته شده نبودند و آن ها نمی خواستند آن ها را فاش کنند. با تمام این بحث ها، NIST [۱] در سال ۱۹۹۴ دوباره صلاحیت DES را برای استفاده دولتی تا ۵ سال دیگر در زمینه هایی به جز اطلاعات رده- بندی شده یا سری [۲] تایید کرد.

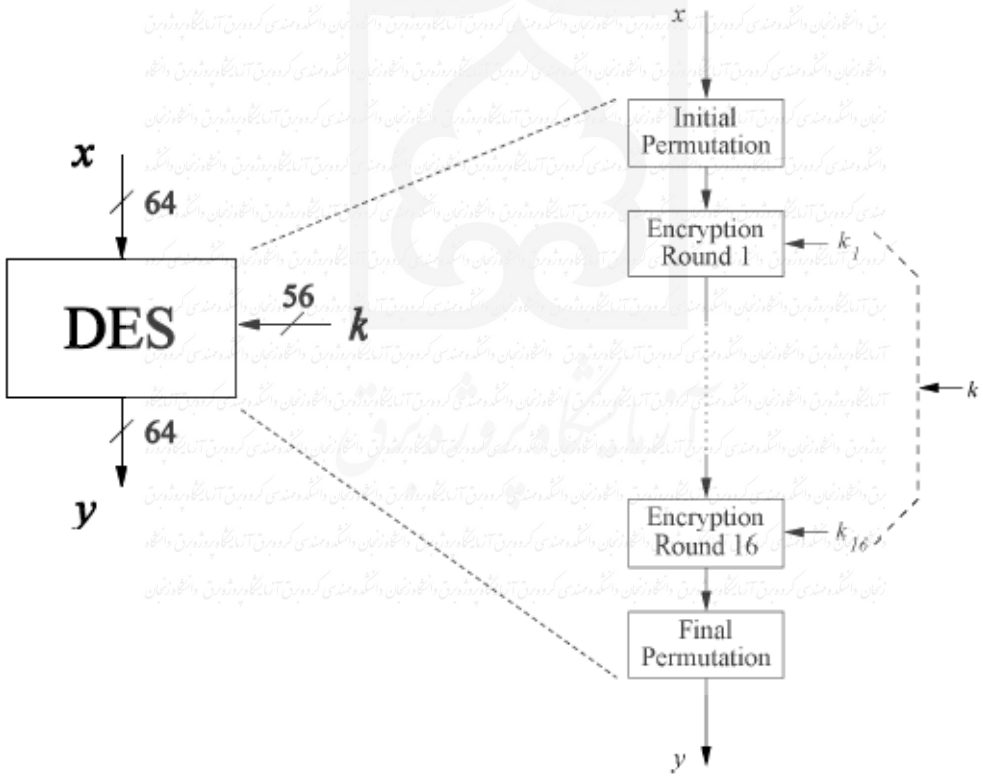
DES تنها روش رمزنگاری کلید متقارن نیست. تعداد زیادی روش های دیگر، هر کدام با سطوح پیچیدگی متفاوت وجود دارند. این نوع رمزنگاری ها شامل : IDEA, RC4, RC5, RC6 و AES می باشند. AES یک الگوریتم مهم است و از ابتدا به هدف جایگزینی DES، به عنوان الگوریتم استاندارد برای رمزنگاری اطلاعات طبقه بندی نشده مطرح گردید. با این حال در سال ۲۰۰۳، AES با کلیدهایی به طول ۱۹۲بیت و ۲۵۶بیت برای حفاظت اطلاعاتی تا حد فوق محرمانه [۳] نیز حائز صلاحیت شناخته شد. تا به امروز AES توانسته در مقابل تمام حمله هایی که به آن شده مقاومت کند، اما تحقیق و بررسی همچنان ادامه دارد تا معلوم گردد این الگوریتم تا چه زمانی می تواند مقاوم باقی بماند.

[۱] National Institute of Standards and Technology
[۲] classified
[۳] top secret

فصل ۲

عملکرد داخلی الگوریتم DES

DES و اکثر رمزنگاری های کلید متقارن دیگر براساس شیوه ای از رمزنگاری به نام Feistel block cipher پایه گذاری شده اند. Feistel block cipher یک بلوک رمز است[۱] که در اوایل دهه ۱۹۷۰، توسط Horst Feistel محقق شرکت IBM طراحی گردید و دارای تعدادی راند[۲] است و هر راند نیز شامل تعدادی تغییر مکان بیت ها[۳]، جاینشینی های غیرخطی[۴] و عمل XOR می باشد. امروزه بیشتر رمزنگاری های کلید متقارن بر اساس همین ساختار طراحی می گردند که به عنوان feistel network شناخته می شود.

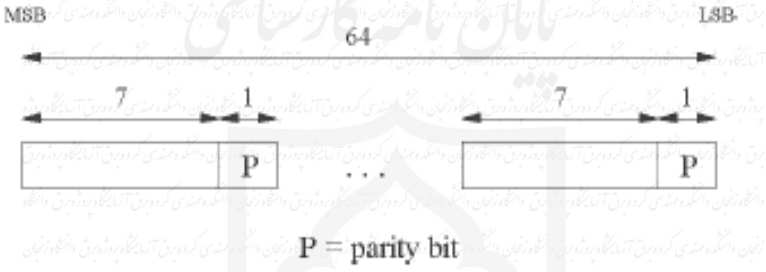


شکل ۲ : شمای کلی رمزنگاری الگوریتم DES

[۱] Block cipher
[۲] round
[۳] bit-shifting
[۴] S-box

همانند بیشتر طرح های رمزنگاری، DES دو ورودی می پذیرد، اول پیام[۱]، برای اینکه عملیات رمزنگاری روی آن انجام گیرد و دوم کلید رمز[۲]. DES یک رمزنگاری کلید متقارن با بلوک رمز به طول ۶۴بیت [۳] است، به این دلیل که از یک کلید مشابه برای رمزنگاری و رمزگشایی استفاده می کند و در هر زمان عملیات را تنها روی یک بلوک اطلاعات[۴] به طول ۶۴بیت انجام می دهد . طول کلید استفاده شده ۵۶ بیت است، با این حال ۶۴ بیت یا ۸ بایت به عنوان کلید ورودی دریافت می شود.

کم ارزش ترین بیت در هر بایت به عنوان بیت تعادل[۵] مورد استفاده قرار می گیرد، که می تواند به صورت دلخواه انتخاب شود زیرا تاثیری در افزایش امنیت رمزنگاری ندارد. DES از بیت تعادل فرد[۶] استفاده می کند. تمام بلوک ها از چپ به راست شماره گذاری شده اند، در نتیجه بیت هشتم هر بایت، بیت تعادل آن است.



شکل ۳: بیت های کلید ورودی

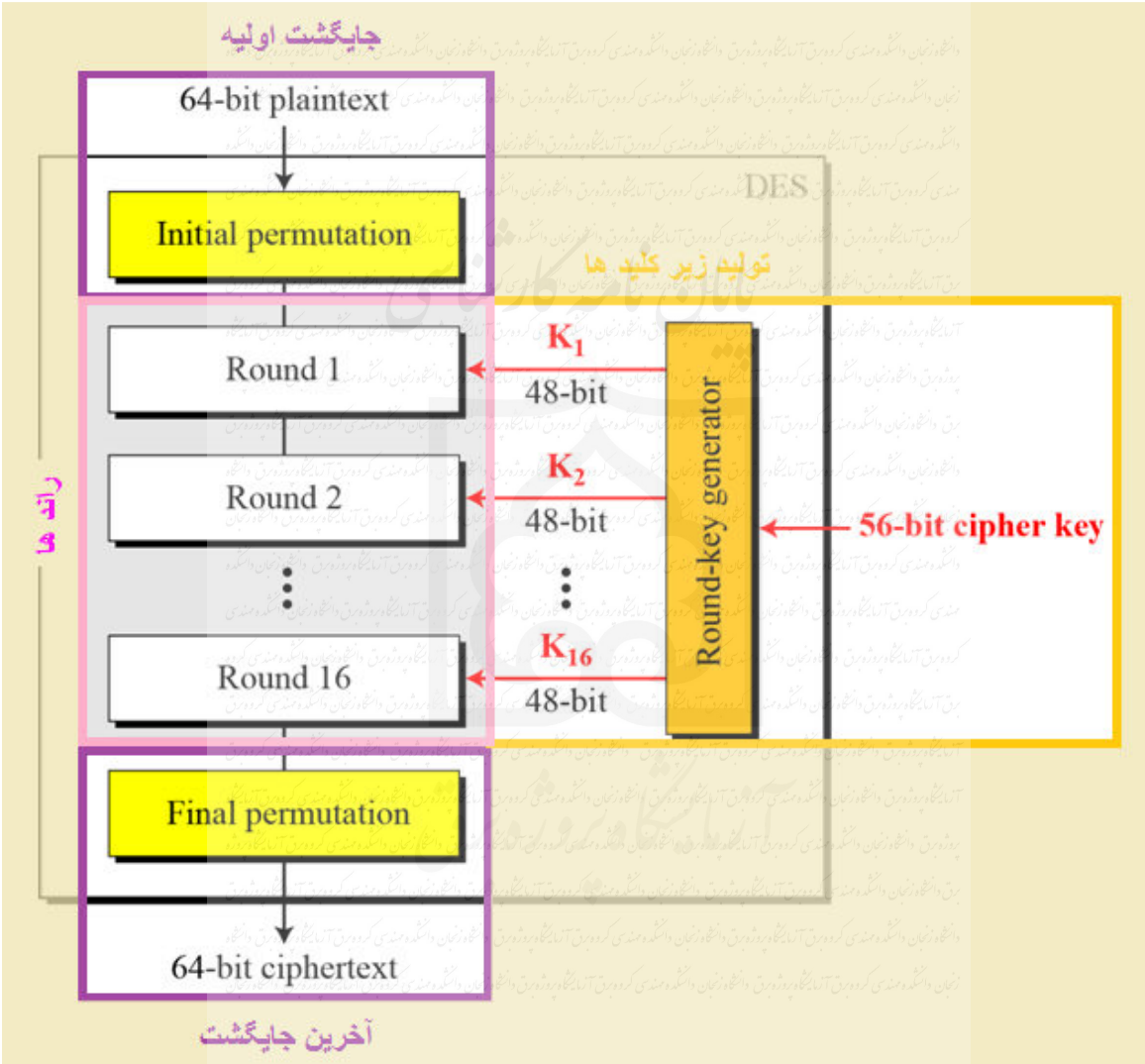
زمانی که پیام ورودی دریافت می شود تا تبدیل به پیام کدگذاری شده گردد، ابتدا به بلوک هایی به طول ۶۴ بیت تقسیم می شود. اگر تعداد بیت های پیام ورودی مضربی از ۶۴ نباشند در این صورت بلوک آخر جبران سازی[۷] می گردد. تعداد زیادی جاینشینی و جایگشت به کار گرفته می شوند تا عملیات رمزگشایی رمزگشایی را سخت تر کنند. با این حال پذیرفته شده که اولین و آخرین جایگشت تاثیری ناچیزی از جنبه امنیتی دارند و در بسیاری از نرم افزارها نیز به خودی خود حذف می شوند.

---

[۱] plaintext
[۲] key
[۳] 64bit block cipher
[۴] پیام یا پیام کدگذاري شده
[۵] parity bit
[۶] Odd parity
[۷] padded

الگوریتم DES را با بررسی سه بخش اصلی آن شرح می دهیم :

- بخش جایگشت اولیه[۱] و آخرین جایگشت[۲]
- بخش راند[۳] ها
- بخش تولید زیرکلید ها[۴]



شکل ۴: سه بخش اصلی الگوریتم DES

[۱] initial permutation
[۲] final permutation
[۳] round
[۴] Round-key generator

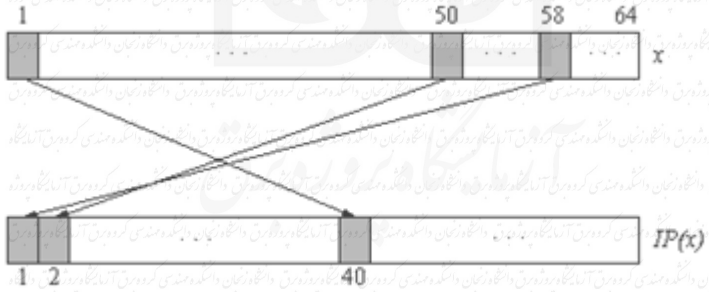
۲-۱ جایگشت اولیه و آخرین جایگشت

DES در ابتدا، جایگشت اولیه را روی بلوک ۶۴ بیتی پیام ورودی انجام می دهد. جایگشت اولیه، آرایش بیت های پیام ورودی را عوض می کند تا پیام جایگردانی شده بدست آید. بدین منظور جدول جایگشت IP مورد استفاده قرار می گیرد .

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

شکل ۵: جدول جایگشت IP

اولین عدد جدول فوق ۵۸ است، این بدان معناست که ۵۸امین بیت پیام ورودی به اولین بیت پیام جایگردانی شده تبدیل می گردد. چگونگی عملکرد جایگشت اولیه در شکل زیر نمایان است :



شکل ۶: طریقه عملکرد جدول جایگشت IP

به عنوان مثال، اگر ورودی جدول جایگشت IP بصورت hexadecimal مقدار زیر باشد :

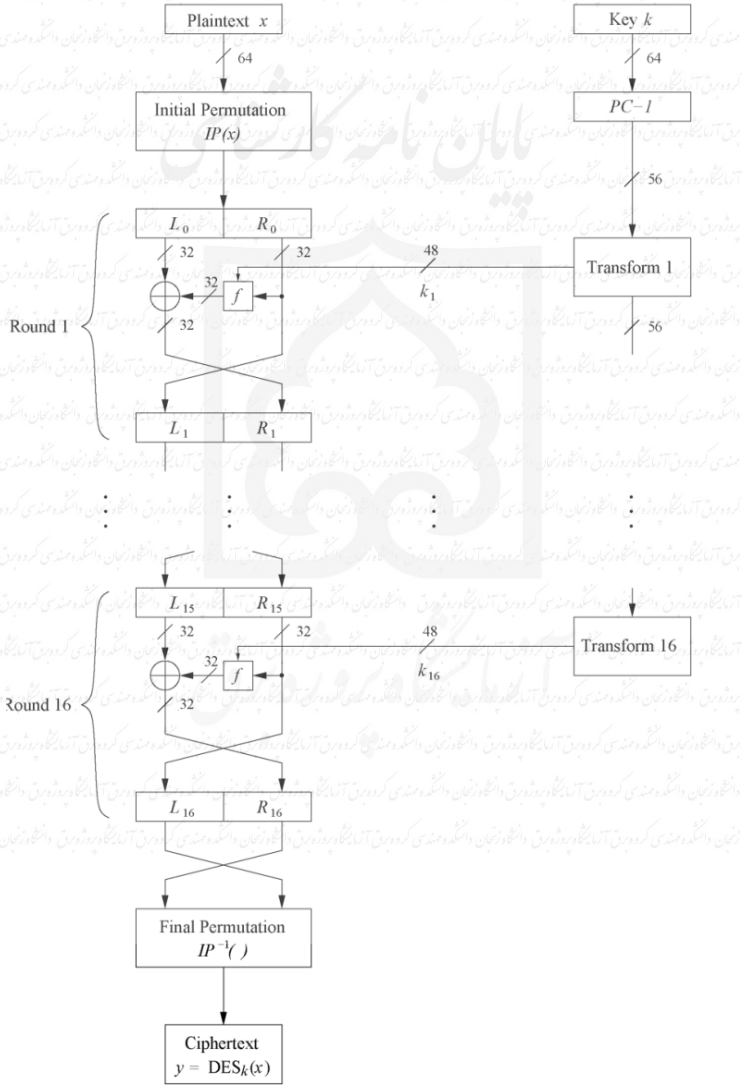
$$x = 0x0002\ 0000\ 0000\ 0001$$

مشاهده می شود تنها بیت های ۱۵ ام و ۶۴ ام مقدار ’1‘ دارند، و بقیه بیت ها ’0‘ هستند. در جایگشت اولیه، طبق جدول جایگشت IP، بیت ۱۵ام تبدیل به بیت ۶۳ام و بیت ۶۴ام تبدیل به بیت ۲۵ام می شود،

بنابراین خروجی این جایگشت برابر با مقدار زیر می گردد:

$$IP(x) = \ 0x0000\ 0080\ 0000\ 0002$$

خروجی جدول جایگشت IP به دو زیربلوک ۳۲بیتی با نام های R_0 و L_0 تقسیم شده و این زیر بلوک ها وارد قسمتی می گردند که راند اول نامیده می شود. تعداد راند ها در الگوریتم DES، ۱۶ عدد است و تمام راند ها مشابه یکدیگرند. در پایان شانزدهمین راند، جای زیربلوک های ۳۲بیتی R_{16} و L_{16} عوض شده و سپس در کنار یکدیگر قرار می گیرند تا آخرین جایگشت که دقیقا معکوس جایگشت اولیه است، روی $R_{16}L_{16}$ انجام شود و خروجی آن که پیام کدگذاری شده می باشد، تولید گردد.



شکل ۷: جایگشت اولیه و آخرین جایگشت در الگوریتم DES

دانشجویان محترم:

جهت دسترسی به متن کامل پایان نامه‌ها به کتابخانه دانشکده مهندسی و یا آزمایشگاه پروژه گروه برق مراجعه فرمایید.