난수 XOR 특성을 활용한 독자 SPN 구조 암호 알고리즘 설계

이원준[1], 장종욱*[1]

[1]동의대학교

Designe of an Independent Symmetric Cipher algorithm which apply characteristic of random number XOR operation

Won-Jun Lee[1] · Jong-Wook Jang[2*]

[1]Dong-eui University

E-mail : dnjswns648@naver.com / jwjang@deu.ac.kr

요 약

정보통신 기술이 발달하면서 동시에 암호화 기술의 중요성이 굉장히 중요해졌다. 암호기술이 없다면 어마어마한 정보가 타인에게 노출되게 된다. 이에 암호기술의 중요성을 인식하고, 암호화에서 유용한 XOR 연산의 특성을 독자적으로 구현한 P-Box와 결합하여 단순 비트 교환을 개선한 대칭 암호의 구조를 구현하고자 하였다. 구현한 암호 알고리즘이 좋은 성능을 갖기 위해 Shannon이 제시했던 혼돈과 확산 특성을 적용하고, 암호 알고리즘 전체 구성요소의 작용을 조화시키려 하였다.

ABSTRACT

With the development of information and communication technology, the importance of encryption technology has become very important. Without cryptography, tremendous information is exposed to others. To this end, we recognized the importance of cryptography, and tried to implement a symmetric cryptography structure that improved the simple bit exchange by combining the characteristics of XOR operations useful in cryptography with a uniquely implemented P-Box. In order for the implemented cryptographic algorithm to have good performance, we applied the Confusion and Diffusion characteristics proposed by Shannon and tried to harmonize the functions of all components of the cryptographic algorithm.

키워드

S-Box, P-Box, SPN, Key Generator

Ⅰ. 서 론

난수와 XOR 연산을 취하면 연산 결과 패턴이 사라진다. 그리고 다시 한번 같은 난수와 XOR 연산을 수행하면 원래대로 돌아온다. 이와 같은 특징은 암호 알고리즘에서 상당히 유용하게 작용한다. 랜덤한 비트열로 만들어주며, 동시에 복호화도 보장하기 때문이다. 블록 암호 구조 중 AES가 채택한 Rijdeal 알고리즘은 SPN 구조를 이용한다. SPN 구조는 반복적으로 XOR 연산을 취하지만 S-Box나 P-Box 내에서는 XOR 연산을 이용하지 않는다. 이에 XOR 연산의 특성과 속도를 Permutation 과정에서 적용하여 우수한 특성을 지니는 블록 암호 알고리즘을 설계하고자 하였다.

이 논문에서는 독자적으로 설계한 P-Box 알고리즘을 사용한다. 또 라운드 수를 반복할수록 특성이 강해지는 암호를 구현하고자 하였다. 이 암호 알고리즘을 통해 평문의 기밀성을 유지할 수 있다.

Ⅱ. 암호 알고리즘 사양

구현할 암호 알고리즘은 대칭 암호 알고리즘 중에서도 블록 암호 알고리즘으로 SPN 구조를 채택하며, 총 16라운드의 길이를 가진다. 평문 블록의 길이는 128-비트로 하도록 한다. 키의 길이 또한 128-비트로 한다. 본 암호 알고리즘은 128-비트 평문과 128-비트 키를 입력받아 128-비트의 암호문 블록을 생성해낸다.

Ⅲ. 라운드 구조

암호 알고리즘은 한 라운드에서 Sub-Bytes, Spaghetti Pipe, Add-Round Key라는 총 3개의 단계를 거친다. Sub-Bytes는 S-Box에 해당하며, 한 바이트 단위로 입력받아 0~255 범위의 인덱스를 가지는 치환표를 이용해 변환된 결과를 출력한다. Spaghetti Pipe는 P-Box에 해당하며, Spaghetti Noodle, Beat-Boiling이라는 처리를 수행하고, 동전을 던져 얻어낸 128-비트 길이의 비트열과 XOR 연산을 수행한다. 이후 한 번 더 Spaghetti Noddle과 Beat-Boiling을 수행한다. 이를 통해 S-Box의 입력 단위인 바이트를 구성하는 모든 비트를 섞는다. 마지막으로 라운드 키와 XOR 연산을 취한다. 아래는 암호 알고리즘의 한 라운드 구조를 나타내는 그림이다.

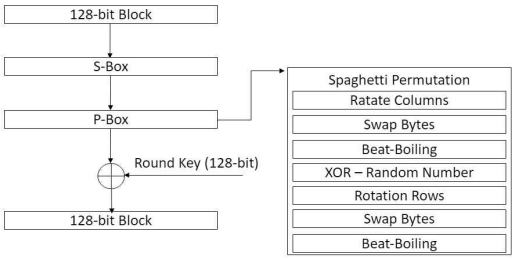

그림 1. 암호 알고리즘 1라운드 구조

.
.

Ⅳ. S-Box Design

구현할 S-Box는 모듈로 곱셈에서의 곱셈의 역원을 계산해 이를 바탕으로 행렬 곱셈을 수행한다. S-Box는 256가지의 크기를 갖는 치환표이며, 행과 열의 순서가 바로 적용할 때 기준이 되는 첨자다. S-Box는 GF(2^8) 필드 내의 유한체 연산으로 수정된 유클리드 알고리즘을 이용해 계산한 모듈로 곱셈의 역원과 행렬 곱을 이용해 구해낸다. 다음 그림은 구현된 S-Box를 정리한 그림이다.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	5F	C7	41	C0	4C	80	81	2B	56	32	01	90	CD	69	98	45
1	AC	3C	AA	C3	CC	E2	21	4E	9B	6F	1C	99	FF	1F	44	2E
2	59	CB	78	15	55	D1	49	AF	57	1B	0B	4A	42	EE	52	73
3	37	29	10	28	F6	7D	33	FC	31	74	3E	0C	88	11	5C	43
4	B2	38	97	F4	F0	85	E4	1E	64	D0	6D	58	92	08	91	3F
5	60	6A	36	8B	16	CF	5A	E0	84	A8	DD	D2	A4	A7	E6	A9
6	6E	89	9C	C8	20	A3	50	3B	23	CA	34	C1	66	D7	F9	9E
7	62	18	26	B9	7C	FB	D6	4B	DF	5D	EC	12	76	C9	48	6C
8	AB	B7	70	8F	E1	79	27	71	2F	19	C5	D3	07	E3	F2	40
9	06	3A	A1	EF	14	94	7E	BE	25	8D	DE	77	ED	E5	B0	E8
A	0E	54	D4	B4	A2	35	17	E7	2C	68	9F	04	7A	7B	0F	C2
B	09	AE	51	83	BB	6B	A5	9A	87	65	4F	8C	03	C4	9D	B6
C	DC	C6	13	95	F7	1D	00	2D	8E	EB	47	F1	A0	DB	B8	BC
D	46	D5	5B	05	A6	30	4D	B5	02	53	61	DA	3D	2A	F3	FD
E	0A	CE	FE	72	82	EA	BD	7F	F8	F5	39	8A	AD	63	96	E9
F	BF	1A	BA	B3	D9	0D	24	86	22	FA	93	B1	5E	67	D8	75

그림 3 S-Box 치환표

Ⅴ. P-Box Design

SPN 구조에서 S-Box도 중요하지만 가장 중요한 부분은 바로 P-Box이다. P-Box는 블록 내 모든 비트를 섞어주는 역할을 한다. 이런 역할을 수 라운드에 거쳐 수행하기 때문에 암호문 블록의 기밀성이 보장되며, 비트 하나의 변화가 많은 비트 변화에 영향을 주게 된다.

본 연구에서는 Spaghetti Pipe라는 독자적인 P-Box를 구현하도록 한다. P-Box의 구조는 아래와 같다.

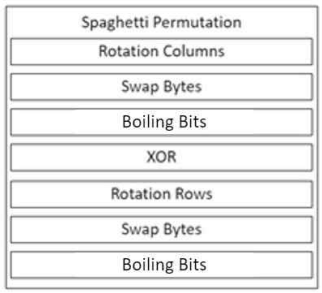

그림 4 The structure of a Spaghetti Pipe

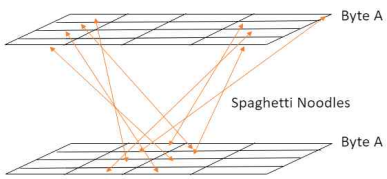

그림 2 Spaghetti Noodle - Swap Byte

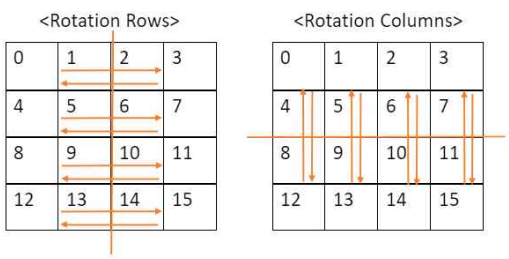

그림 5 Rotation Rows, Rotation Columns

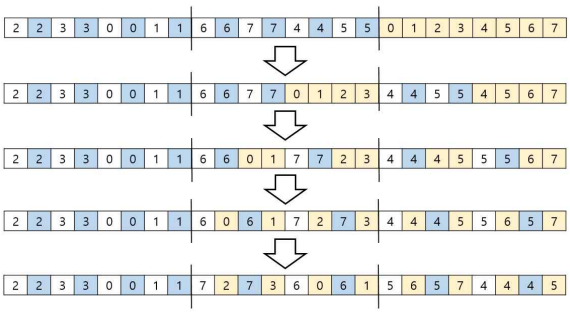

그림 6 3 Bytes Boiling Bits

Ⅵ. Key Generator Design

암호 알고리즘 내에서 S-Box와 P-Box 만큼 중요한 요소가 바로 Key Generator이다. 대칭 암호 알고리즘에서 키는 무조건 숨겨져야 하지만, 암호문을 보고 키를 추측할 수도 있으므로 키도 충분한 처리를 거쳐야 한다.

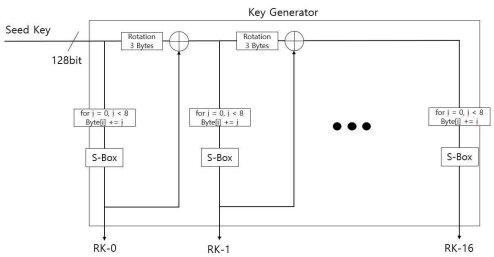

그림 7 Key Generator

Ⅴ. 결 론

성공적으로 암호 알고리즘을 구현했고, 이를 통해 1비트 차이가 암호문 전체에 영향을 줄 수 있다는 사실을 확인할 수 있었다. 이러한 사실만으로는 본 연구의 암호 알고리즘이 '안전하다'고 말할 수는 없지만, 블록 암호 알고리즘의 특성과 확산, 혼돈 특성이 존재하는 것처럼 보이므로, 암호 알고리즘으로써 기본 조건을 충족한다고 생각한다.

Acknowledgement

이 논문은 **** 지원에 의함.

References

[1] Wikipedia. Adanced Encryption Standard(Rijndeal). Available : https://ko.wikipedia.org/wiki/%EA%B3%A0%EA%B8%89_%EC%95%94%ED%98%B8%ED%99%94_%ED%91%9C%EC%A4%80

[2] Hiroshi Yuki, Introduction to Information Protection,
Jae Kwang Lee, Lee, Tae-il Jeon, Jaeshin Shin, Seoul, Korea, Infinity Books, 95~197, 2017,

[3] J. G. Proakis, *Digital Communications,* 4th ed. New York, NY: McGraw-Hill, 1993.