

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2010-0124021 (43) 공개일자 2010년11월26일
(51) Int. Cl. <i>H04L 9/06</i> (2006.01) (21) 출원번호 10-2009-0043052 (22) 출원일자 2009년05월18일 심사청구일자 2009년05월18일	(71) 출원인 부경대학교 산학협력단 부산 남구 용당동 산 100번지 부경대학교내 (72) 발명자 김길호 부산 서구 초장동 110-2번지 8/2 김원희 부산광역시 남구 대연동 891-7 장미하우스 가-106 (뒷면에 계속)	

전체 청구항 수 : 총 6 항

(54) 대칭단을 이용한 알씨6 알고리즘의 암호/복호 과정 동일화 설계 방법

(57) 요약

암호/복호 알고리즘이 서로 다른 블록 암호 RC6를 간단한 논리 연산과 회전 연산을 사용한 대칭단의 삽입으로 암호/복호 과정 동일화 설계방법에 관한 것이다. 본 발명은 RC6의 전체 r라운드 중에서 1라운드부터 r/2라운드까지의 r/2라운드의 암호 과정을 진행하는 암호단, xor연산단, 회전(Rotate) 연산단, and 연산단, or 연산단, 연접 연산단으로 구성되어 1번만 수행하는 대칭단, (r/2)+1라운드부터 r라운드까지 r/2라운드의 복호 과정을 진행하는 복호단을 포함한다.

대표도



(72) 발명자

조경연

부산시 해운대구 좌동 1271번지 신성아파트
104-1402

김종남

부산광역시 남구 용호동 197 GS하이츠자이
308-2801

이광윤

부산광역시 연제구 연산9동 203-10 한솔파크 104동
903호

이강석

부산시 연제구 연산9동 243-18 LG아파트 120-1803
호

박지규

부산 북구 덕천2동 주공아파트 106-907호

이 발명을 지원한 국가연구개발사업

과제고유번호 000303220108

부처명 중소기업청

연구관리전문기관

연구사업명 산학공동기술개발지원사업

연구과제명 디지털 콘텐츠의 불법 유출 방지를 위한 H.26와 SEED 기반의 실시간 비디오 워터마킹 및
암호화 시스템 개발

기여율

주관기관 부경대학교 산학협력단

연구기간 2008년 07월 01일 ~ 2009년 06월 30일

특허청구의 범위

청구항 1

암호단, 대칭단, 복호단으로 구성되는 것을 특징으로 하는 RC6 알고리즘의 암호/복호 과정 동일화 설계 방법.

청구항 2

제 1항에 있어서 암호단은 전체 r 번의 진행 라운드 중에서 1라운드부터 $r/2$ 라운드까지 전체의 $r/2$ 횟수의 라운드만 수행하는 것을 특징으로 하는 RC6 알고리즘의 암호/복호 과정 동일화 설계 방법.

청구항 3

제 1항에 있어서 대칭단은 1번만 수행하는 것을 특징으로 하는 RC6 알고리즘의 암호/복호 과정 동일화 설계 방법.

청구항 4

제 1항에 있어서 복호단은 전체 r 번의 진행 라운드 중에서 $(r/2)+1$ 라운드부터 r 라운드까지 전체의 $r/2$ 횟수의 라운드만 수행하는 것을 특징으로 하는 RC6 알고리즘의 암호/복호 과정 동일화 설계 방법.

청구항 5

제 3항에 있어서 대칭단은 xor 연산단, rotate 연산단, and 연산단, or 연산단, 연접 연산단으로 구성되는 것을 특징으로 하는 RC6 알고리즘의 암호/복호 과정 동일화 설계 방법.

청구항 6

제 5항에 있어서 대칭단의 결과가 아래의 [수학식 1]에 의하여 산출되는 것을 특징으로 하는 RC6 알고리즘의 암호/복호 과정 동일화 설계 방법.

수학식 1

도 1의 대칭단 내의 연산을 수학식으로 표기하면 다음과 같으며, 수학식에 사용된 연산자는 다음과 같다.

$\wedge$: 비트별 xor 연산자.

$\lll$: 왼쪽 rotate 연산자.

$\&$: 비트별 and 연산자.

$|$: 비트별 or 연산자.

$||$: 연접 연산자.

$$\begin{aligned} B' &= (b(A, k[22]) \lll 12) \wedge B \\ A' &= (b(B', k[25]) \lll 29) \wedge A \\ D' &= (b(C, k[24]) \lll 12) \wedge D \\ C' &= (b(D', k[23]) \lll 29) \wedge C \end{aligned}$$

여기서 함수 $b(X, Y)$ 는 32비트 워드 2개 X, Y 를 입력 받아 이 X, Y 를 다시 바이트 단위로 나누어 and, or 연산을 교번 수행 하는 함수로서 다음과 같다.

$$X = x_3 \parallel x_2 \parallel x_1 \parallel x_0$$

$$Y = y_3 \parallel y_2 \parallel y_1 \parallel y_0$$

$$b(X, Y) = b((x_3 \& y_3) \parallel (x_2 \mid y_2) \parallel (x_1 \& y_1) \parallel (x_0 \mid y_0))$$

명세서

발명의 상세한 설명

기술분야

[0001] 블록 암호는 크게 Feistel구조와 SPN(Substitution Permutation Network)구조로 분류할 수 있으며, Feistel구조는 암호/복호 알고리즘이 같다는 것이 장점이고 SPN구조는 암호/복호 알고리즘이 다른 특징이 있다. 특히 하드웨어로 암호 시스템을 구현할 때 암호/복호 알고리즘이 다른 SPN 구조는 Feistel 구조보다 하드웨어 면적이 약 2배 정도 증가하는 단점이 있다.

[0002] 변형된 Feistel 구조 암호 알고리즘인 RC6은 새로운 표준 블록 암호 알고리즘을 선정하는 프로젝트인 AES 공모의 최종 후보 알고리즘이었다. RC6은 Feistel 구조로 암호/복호 알고리즘이 다른 특징을 가지고 있다. 이러한 특징은 암호 알고리즘을 하드웨어로 구현할 때 암호/복호 알고리즘이 같은 것보다 면적이 상당히 증가하는 단점이 된다. 이와 같은 RC6의 단점을 보완하는 암호/복호가 동일한 RC6 암호 알고리즘에 관한 발명이다.

배경기술

[0003] 데이터 무결성(Integrity)을 보장하는 수단으로 데이터에 대한 대칭 키(Symmetric key) 암호화가 일반적이다. 대칭 키 암호는 블록 암호(Block cipher)와 스트림 암호(Stream cipher)로 나눌 수 있으며, 블록 암호는 많은 알고리즘들이 개발 되었고, 특히 미국의 AES(Advanced Encryption Standard) 선정 프로젝트, 유럽의 NESSIE(New European Schemes for Signatures, Integrity, and Encryption) 프로젝트, 일본의 CRYPTREC(Cryptography Research and Evaluation Committees) 프로젝트 등 선진 각국들은 자국의 정보보호를 위한 암호 알고리즘을 공모를 통해 선정하고 있다. 우리나라는 자체 개발한 SEED와 ARIA를 표준 128비 블록 암호로 제정하였다. 초기 블록 암호는 소프트웨어로 구현되었지만 최근에는 빠른 정보처리를 위한 하드웨어 구현에 많은 연구가 진행 중이다.

[0004] RC6은 변형된 Feistel 구조의 암호 알고리즘으로 RC6-w/r/b로 표기한다. 여기서 w는 워드의 크기로 32비트이며, r은 라운드 수로 블록의 크기가 128비트인 경우 20라운드 이고, b는 암호화 키의 바이트 수로 16바이트이다.

[0005] RC6에서 사용된 연산자는 다음과 같다.

- [0006] $A + B$: 정수덧셈 mod 2^w
- [0007] $A - B$: 정수뺄셈 mod 2^w
- [0008] $A \wedge B$: 워드단위 비트 별 xor 연산
- [0009] $A * B$: 정수곱셈 mod 2^w
- [0010] $A \lll B$: B의 $\log w$ 만큼 A를 왼쪽 회전연산
- [0011] $A \ggg B$: B의 $\log w$ 만큼 A를 오른쪽 회전연산
- [0012] $(A, B, C, D) = (B, C, D, A)$: 워드단위 할당연산
- [0013] RC6의 암호/복호는 32비트 워드단위로 A, B, C, D 4개의 저장 장소에 평문/암호문을 가지고 20라운드를 반복 수행 후 암호문/평문을 만들어 낸다. A, B, C, D 4개의 워드는 라운드 함수 수행 후 모두 병렬로 워드 단위 왼쪽/오른쪽 회전 연산이 이루어지며, 라운드 함수 수행 전과 후에 화이트닝(whitening) 단계로 라운드 키와 덧셈/뺄셈을 수행한다. 라운드 함수 내의 핵심적인 안전성은 데이터 의존 회전 연산이고, 이 회전 연산 양은 $f(x) = x(2x + 1)$ 의 2차 함수에 고정된 5비트 왼쪽 회전 연산으로 만들어진다.
- [0014] RC6은 암호에 적용된 라운드 키는 복호할 때는 암호의 역순으로 적용한다. 그리고 RC6은 변형된 Feistel 구조로 암호/복호 알고리즘이 서로 다르다.
- [0015]
- ```

B = B + S[0];
D = D + S[1];
for(i=1; i<r; i++)
{
 t = (B*(2B + 1)) log w;
 u = (D*(2D + 1)) log w;
 A = ((A t) u) + S[2i];
 C = ((C u) t) + S[2i+1];
 (A, B, C, D) = (B, C, D, A);
}
A = A + S[2r+2];
C = C + S[2r+3];

```
- [0016] RC6의 암호 알고리즘
- [0017]
- ```

C = C - S[2r+3];
A = A - S[2r+2];
for(i=r; i>=1; i--)
{
    (A, B, C, D) = (D, A, B, C);
    u = (D*(2D + 1)) log w;
    t = (B*(2B + 1)) log w;
    C = ((C - S[2i+1]) t) u;
    A = ((A - S[2i]) u) t;
}
D = D - S[1];
B = B - S[0];
    
```
- [0018] RC6의 복호 알고리즘
- [0019] S는 32비트 라운드 키로 RC6의 키 스케줄링에 의해 미리 만들어 메모리에 저장해 두고 사용한다.

발명의 내용

해결 하고자하는 과제

- [0020] 본 발명의 암호/복호가 동일한 RC6 알고리즘은 기존의 RC6 알고리즘을 변경 없이 그대로 사용하면서 바이트단위 논리 연산과 고정된 회전 연산으로 구성된 대칭단의 삽입만으로 암호/복호가 동일한 RC6 알고리즘을 만든다. 이와 같은 대칭단의 구성은 소프트웨어 및 하드웨어 구현이 쉽고 수행 속도도 빠르다. 그리고 제안하는 암호/복호가 동일한 RC6 알고리즘은 전체 진행 라운드의 1/2은 암호 알고리즘을 수행하고, 나머지 1/2은 복호 알고리즘을 수행하면서 중간에 대칭단을 삽입하여 암호/복호 알고리즘이 동일한 개선된 RC6를 만든다.
- [0021] 대칭단의 설계목표는 다음과 같다.
- [0022] 암호/복호가 동일한 RC6를 만들 것.
- [0023] 동일한 라운드의 적용에서 대칭단의 삽입으로 불규칙성을 통해서 RC6의 안전성을 향상시킬 것.
- [0024] 소프트웨어 및 하드웨어 구현이 쉬울 것.
- [0025] 기존의 RC6와 소프트웨어 및 하드웨어 수행속도에서 큰 차이가 없을 것.

과제 해결수단

- [0026] 본 발명은 암호/복호를 동일하게 구성하는 개선된 RC6 알고리즘을 개발하는 것으로 크게 3단계로 진행한다.
- [0027] RC6 암호 알고리즘을 전체 라운드 중 1/2만 수행하는 단계.
- [0028] 대칭단을 적용하는 단계.
- [0029] RC6 복호 알고리즘을 나머지 1/2 라운드를 수행하는 단계로 나눌 수 있다.
- [0030] 이와 같은 진행의 구성은 암호/복호가 동일한 RC6 알고리즘을 만들 수 있다.

효 과

- [0031] PDA나 핸드폰과 같은 한정된 하드웨어 자원을 가지고 있는 장치에서 보안이 필수적인 데이터 통신을 위한 암호 알고리즘의 하드웨어 구현은 암호/복호 알고리즘이 다를 경우 암호/복호 알고리즘이 같은 경우보다 하드웨어 면적이 2배 정도 증가하게 된다. 이에 본 발명으로 기존의 암호/복호가 다른 RC6를 암호/복호가 같은 알고리즘으로 구현 했 으며, 기존의 RC6와 유사한 안전성을 유지하면서 암호/복호가 동일하기 때문에 하드웨어 구성이 간단하고 쉽게 구현 할 수 있다. 또한 스마트카드 및 능동형 RFID 태그와 같은 제한된 하드웨어 및 소프트웨어 환경에서도 쉽게 구현 가능하다.
- [0032] 본 발명의 핵심인 대칭단은 간단한 논리연산으로 구성되어 소프트웨어 및 하드웨어 수행속도가 빠르다. 다시 말 해 기존의 RC6와 소프트웨어 수행속도 테스트에서 거의 같은 수행속도를 보여주고 있으며 이는 현재 RC6 알고리즘을 적용하고 있는 암호 시스템에 그대로 적용이 가능하다.
- [0033] 그리고 RC6 알고리즘이외의 다른 SPN구조의 블록 암호 알고리즘들과 암호/복호 알고리즘이 다른 유사한 구조의 암호 알고리즘에 본 발명의 대칭단 구조를 삽입해서 암호/복호 알고리즘을 동일하게 구현할 수 있고, 새로운 암호/복호 알고리즘이 같은 블록 암호 설계 및 개발에도 유용한 아이디어로 사용될 것이다.

발명의 실시를 위한 구체적인 내용

- [0034] 본 발명에서 대칭단 구조를 RC6 알고리즘에 적용할 때 대표도의 100 RC6 암호 알고리즘과 300 RC6 복호 알고리즘은 변경 없이 그대로 사용한다. 그러나 전체 진행 라운드에서 1/2라운드만 암호 알고리즘을 사용하고 나머지 1/2라운드는 복호 알고리즘을 사용하고, 암호 알고리즘에서 복호 알고리즘으로 바뀌는 중간에 대칭단 연산이 삽입된다(대표도 참조).
- [0035] RC6의 암호 알고리즘 구성은 모든 라운드에 동일한 연산이 반복되면 수학식이 단순해져서 선형 공격에 취약하다. 그래서 1 라운드 수행 전에 도 1의 21, 41과 같이 라운드 키와 덧셈을 수행 후 라운드 함수를 적용하

고 10라운드 암호과정을 수행한다. 10라운드 암호과정은 RC6의 암호과정을 변경없이 그대로 수행하고 대표도 200의 대칭단을 1번 수행한다.

[0036] 상세한 대칭단 수행 과정은 도 1에 나타났다. 먼저 도 1의 210은 암호과정 수행 후 업데이터 된 32비트 B와 라운드 S[22]을 입력으로 받아 32비트 워드를 바이트 단위로 나눈 다음 and, or 연산을 교번 수행한다. 그리고 도 1의 211은 210의 수행 후 결과 값에 12비트 왼쪽 회전연산을 수행하고 220은 211의 결과 값과 암호 수행 후 업데이터 된 32비트 C와 xor연산을 수행한다. 220의 결과 값은 최종적으로 대칭단의 출력으로 나가면서 다시 221의 입력으로 사용된다. 221은 220의 결과값과 라운드 키 S[25]를 입력 받아 워드를 바이트 단위로 나눈 다음 and, or 연산을 교번 수행한다. 221을 수행 후 결과 값을 222의 29비트 회전연산을 수행하고 223은 222의 결과 값과 암호 과정후 업데이터 된 32비트 B와 xor연산을 수행한 후 32비트 대칭단 출력이 된다.

[0037] 230은 암호과정 수행 후 업데이터 된 32비트 D와 라운드 S[24]을 입력으로 받아 32비트 워드를 바이트 단위로 나눈 다음 and, or 연산을 교번 수행한다. 그리고 도 1의 231은 230의 수행 후 결과 값에 29비트 왼쪽 회전연산을 수행하고 240은 231의 결과 값과 암호 수행 후 업데이터 된 32비트 A와 xor연산을 수행한다. 240의 결과 값은 최종적으로 대칭단의 출력으로 나가면서 다시 241의 입력으로 사용된다. 241은 240의 결과값과 라운드 키 S[23]를 입력 받아 워드를 바이트 단위로 나눈 다음 and, or 연산을 교번 수행한다. 241을 수행 후 결과 값을 242의 12비트 회전연산을 수행하고 243은 242의 결과 값과 암호 과정후 업데이터 된 32비트 D와 xor연산을 수행한 후 대칭단의 출력 32비트가 된다.

[0038] 대칭단 수행 후 대표도의 300의 RC6 10라운드 복호 알고리즘을 수행하고 복호 알고리즘은 RC6의 복호 알고리즘을 변경없이 그대로 사용한다.

[0039] 복호과정 과정 수행 후 마지막으로 도 1의 61, 81과 같이 라운드 키 뺄셈을 수행 한 후 최종적인 암호문을 만든다. 복호는 위의 과정을 그대로 수행하고 단지 대칭단을 역순으로 적용하고 라운드 키 또한 암호과정의 역순으로 적용하면 암호문이 평문으로 변환 된다. 본 발명에서 제안한 알고리즘의 키 스케줄링은 RC6의 키 스케줄링을 그대로 사용하고, 단지 대칭단에서 사용하는 32비트 라운드 4개를 추가하여 총 32비트 48개의 라운드 키를 사용한다.

도면의 간단한 설명

[0040] 대표도는 본 발명의 전체 알고리즘 간소화된 진행도.

[0041] 도 1은 대표도의 상세도로 대표도 100 RC6 암호 알고리즘 진행과정으로 RC6 전체 라운드 중에서 1/2 라운드만 진행하고, 다음으로 대표도 200 대칭단 진행과정으로 한번만 수행하고, 마지막으로 대표도 300 RC6 복호 알고리즘 진행과정으로 RC6 전체 라운드 중에서 나머지 1/2 라운드만 진행한다.

[0042] 도면의 주요 부분에 대한 부호 설명.

[0043] 10 : 평문 첫 번째 워드.

[0044] 20 : 평문 두 번째 워드.

[0045] 21 : 평문 두 번째 워드와 첫 번째 라운드 키와 덧셈 수행부.

[0046] 30 : 평문 세 번째 워드.

[0047] 40 : 평문 네 번째 워드.

[0048] 41 : 평문 네 번째 워드와 두 번째 라운드 키와 덧셈 수행부.

[0049] 50 : 암호문 첫 번째 워드.

[0050] 60 : 암호문 두 번째 워드.

[0051] 61 : 복호과정 수행 후 업데이터 된 32비트 B와 46 번째 라운드 키와 뺄셈 수행부.

[0052] 70 : 암호문 세 번째 워드.

[0053] 80 : 암호문 네 번째 워드.

[0054] 81 : 복호과정 수행 후 업데이터 된 32비트 D와 마지막 라운드 키와 뺄셈 수행부.

- [0055] 100 : RC6 암호 알고리즘 10라운드 수행부.
- [0056] 110 : 32비트 A와 120 f함수 수행결과와 xor 연산 수행부.
- [0057] 111 : 110 xor 연산 후 140 f함수 수행결과와 회전연산 수행부.
- [0058] 112 : 111 회전 연산 후 라운드 키와 덧셈 연산 수행부.
- [0059] 120 : 21 덧셈 연산 후 이차함수 $f = x(2x + 1)$ 의 f함수 수행부.
- [0060] 130 : 32비트 C와 140 f함수 수행결과와 xor 연산 수행부.
- [0061] 131 : 130 xor 연산 후 120 f함수 수행결과와 회전연산 수행부.
- [0062] 132 : 131 회전 연산 후 라운드 키와 덧셈 연산 수행부.
- [0063] 140 : 41 덧셈 연산 후 이차함수 $f = x(2x + 1)$ 의 f함수 수행부.
- [0064] 200 : 대칭단 적용부.
- [0065] 210 : 암호 10라운드 후 업데이터 된 B와 라운드 키 k[22]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.
- [0066] 211 : 210 과정을 수행 후 12비트 왼쪽 rotate 연산 수행부.
- [0067] 220 : 211 과정을 수행 후 암호 10라운드 후 업데이터 된 32비트 C와 xor 연산 수행부.
- [0068] 221 : 220 과정을 수행 후 라운드 키 k[25]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.
- [0069] 222 : 221 과정을 수행 후 29비트 왼쪽 rotate 연산 수행부.
- [0070] 223 : 222 과정을 수행 후 암호 10라운드 후 업데이터 된 32비트 A와 xor 연산 수행부.
- [0071] 230 : 암호 10라운드 후 업데이터 된 D와 라운드 키 k[24]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.
- [0072] 231 : 230 과정을 수행 후 29비트 왼쪽 rotate 연산 수행부.
- [0073] 240 : 231 과정을 수행 후 암호 10라운드 후 업데이터 된 32비트 A와 xor 연산 수행부.
- [0074] 241 : 240 과정을 수행 후 라운드 키 k[23]을 입력 받아 and, or 연산을 바이트 단위로 교번 수행부.
- [0075] 242 : 241 과정을 수행 후 12비트 왼쪽 rotate 연산 수행부.
- [0076] 243 : 242 과정을 수행 후 암호 10라운드 후 업데이터 된 32비트 C와 xor 연산 수행부.
- [0077] 300 : RC6 복호 알고리즘 10라운드 수행부.
- [0078] 310 : 대칭단 적용 후 업데이터 된 32비트 D와 라운드 키 뺄셈 수행부.
- [0079] 311 : 310 뺄셈 수행 후 340 f함수 수행결과와 회전연산 수행부.
- [0080] 312 : 311 회전연산 수행 후 320 f함수 수행결과와 xor 연산 수행부.
- [0081] 320 : 대칭단 수행 후 업데이터 된 32비트 A를 이차함수 $f = x(2x + 1)$ 의 f함수 수행부.
- [0082] 330 : 대칭단 적용 후 업데이터 된 32비트 B와 라운드 키 뺄셈 수행부.
- [0083] 331 : 330 뺄셈 수행 후 320 f함수 수행결과와 회전연산 수행부.
- [0084] 332 : 331 회전연산 수행 후 340 f함수 수행결과와 xor 연산 수행부.
- [0085] 340 : 대칭단 수행 후 업데이터 된 32비트 C를 이차함수 $f = x(2x + 1)$ 의 f함수 수행부.

도면

도면1

