論文
9-2-8

Cellular Automata 기초로 형성된 Stream Cipher - Cellular Automata rule 30을 중심으로 - Completed Stream Cipher by Cellular Automata - About Cellular Automata rule 30 -

남 태 희(Tae Hee Nam)[1)]

요 약

본 논문은 Cellular Automata 기초로 형성된 stream cipher에 대해 원리를 분석하였다. 원래 Cellular Automata는 State, Neighborhood, Transition Rules이라는 단순한 특징을 가지고 복잡하고 다양한 원리를 구현할 수 있다. 즉 Cellular Automata는 transition rule를 이용하여 암호화를 원활하게 처리할 수 있다는 것을 암시하고 있다. 따라서 본 논문에서는 Cellular Automata의 transition rule 30 적용으로 binary pad(key stream)을 생성하고, 암호 분류 중 symmetric key encryption 방식의 stream cipher를 이용하여 encryption 및 decryption의 능력을 실험하였다.

ABSTRACT

In this study, analyzed principle about stream cipher that is formed to Cellular Automata foundation. Cellular Automata can embody complicated and various principle with simple identifying marks that is State, Neighborhood, Transition Rules originally. Cellular Automata is hinting that can handle encipherment smoothly using transition rule. Create binary pad (key stream) by Cellular Automata's transition rule 30 applications in treatise that see therefore, and experimented ability of encryption and decryption because using stream cipher of symmetric key encryption way of password classification.

논문 접수 : 2008. 4. 30.
심사 완료 : 2008. 5. 14.

1) 동주대학 의료기공학과 교수

1. 서론

암호화(encryption)는 인류 역사가 시작되기 전부터 사용되어 온 것으로 알려지고 있다. 그러나 국가가 형성되기 전 비밀리에 보관해야 할 정보가 그리 많지 않아 암호화 방식의 사용이 거의 없었다. 이후 국가가 형성되고, 상업이 발달함에 따라서 개인 및 국가의 이권에 따른 비밀 보전의 필요성이 증대되면서 암호학이 대두되기 시작하였다. 이러한 암호화(encryption)가 작게는 개인의 소중한 비밀을 보호하고 크게는 국가적 중요 정보를 보호하는데 목적이 있다. 암호화에 관련되는 주요 문서는 군사기밀, 정부정책문서, 사업기밀 등이 암호 시스템의 주요 암호화 대상 요인이다. 본 논문은 symmetric key 암호 방식에서 stream cipher의 XOR 연산자를 이용하여 암호화하는 원리를 제시하고자 한다. 특히 암호화에 사용되는 key stream 생성은 1차원 cellular automata의 transition rule 30을 적용하여 평문에 XOR 암호화를 수행함으로서 효과적인 암호화 수행 능력을 검증한다.

2. Cellular Automata 원리

Cellular Automata(CA)의 원리 및 특성은 chaotic system과 artificial life을 연구하는데 주요한 수단으로 간주되고 있으며 동적이고 복잡한 자연현상을 시뮬레이션 하는데 유용한 도구이다. CA 모델은 네 가지의 기본 요소로 구성된다. 첫째는 cellular 공간으로 무한한 다차원의 공간을 동일한 크기의 셀로 분할한 공간을 의미하며, 여러 가지의 정규 다각형으로 분할이 가능하다. cellular automata는 자연의 현상(natural complex systems)을 계산의 형태로 나타내고자 하는 것으로 객체를 상호작용하는 오토마타인 셀들 즉 각 셀은 나름대로의 규칙을 갖는다는 것이고, 이들을 집단으로 보고 이 셀들 사이의 상호작용을 나타내는 것이다. 두 번째 구성요소는 지역 상태로, 이는 주어진 시간에서 각 셀의 상태, 즉 셀이 가지는 값을 의미한다. 세 번째는 이웃 셀(Neighborhood)과의 관계이다. 이웃 셀은 중심 셀 주위에 인접하고 있는 cell들의 집합을 의미하는데, 보통 거리, 방향 혹은 각도에 의해서 결정된다. CA의 마지막 구성요소는 transition rule이다. transition rule은 각 셀이 매 시간별로 어떻게 변화할 지에 대해 rule을 정하는데, 여기에는 이웃 셀들의 구성과 위치가 정의되어 있다. transition rule은 모든 cellular 공간에서 동일하게 적용되는 공통 규칙(universal rule)으로 지역적인 특성을 지니고 있다. 정리하면, CA란 독립된 동적 시스템들이며, 시간과 공간을 이산적으로 다루는 시스템으로 cellular 공간의 기본단위는 하나의 셀로 구성되어 있다. 1차원 CA는 모든 셀들이 선형으로 배열되어 있다. 중요한 것은 국소적 상호작용이 세 개의 셀, 즉 자기 자신과 인접한 두 셀에 의해 이루어지는 3-이웃(3-Neighbourhood) CA이다.

i : 일차원으로 배열되어 있는 각 셀들의 위치
t : 시간

$q_i(t)$: 시간 t에서 i 번째 셀의 상태
$q_i(t+1)$: 시간 $t+1$에서 i 번째 셀의 상태

•	•	•	$q_{i-1}(t)$	$q_i(t)$	$q_{i+1}(t)$	•	•	•

[그림 1] 중간 셀 q_i 상태와 시간 t에 대해 q_{i+1}과 q_{i-1}이 서로 인접된 상태

[Fig. 1] The cell states of the central cell q_i and its two nearest neighbors q_{i-1} and q_{i+1} at the time step t.

3-neighbors CA에 대한 상태 전이 함수(state-transition function)는 수학적 표기로서 다음과 같이 정리할 수 있다.

radius $r=1$ 일 때,

$$q_i(t+1) = f[q_{i-1}(t), q_i(t), q_{i+1}(t)]$$

radius $r = 0$ 일 때,

$$q_i(t+1) = f[q_i(t)]$$

3. Bit stream cipher

stream ciphers는 대칭성 키 암호화 계획들의 중요한 클래스를 구성한다. stream cipher는 사용자의 비밀 key로부터 binary pad(key stream) 생성함수를 이용하여 평문(plaintext)과 bits 크기가 같은 key stream을 생성한다. 평문과 key stream을 bit 단위로 암호화할 수 있다. 이것은 블록 암호 알고리듬에서처럼 padding 과정이 필요 없다. stream cipher의 경우, bit 단위로 암호화하므로 오류발생이 확산되지 않고 처리속도가 빠르다. 단 상대적으로 암호화 및 복호화가 간단하므로 key stream 생성함수를 알아내기 어렵게 하여 해독하기 어렵게 해야 한다. stream cipher는 평문에 연속되는 key stream을 적용시켜 암호화하는 방식이다. keystream 조건은 $k_1k_2k_3\cdots\cdots k_i \in K$이며, 조건 $k \in K$, E_k와 단순치환 암호화(simple substitution cipher)가 되도록 하였다.

Encryption Function : E

$$E_k : P \rightarrow C \quad \text{----------(1)}$$

Decryption Function : D

$$D_k : C \rightarrow P \quad \text{----------(2)}$$

여기서 함수 E_k와 D_k은 조건 $p \in P$와 $k \in K$로 정의되며 수식 다음과 같이 정리된다.

$$D_k(E_k(p)) = p \quad \text{----------(3)}$$

평문은 $p_1p_2p_3\cdots p_i$이며 $k_1k_2k_3\cdots k_i$가 K로부터 keystream이도록 하였다.

즉 평문 $P = p_1\, p_2 \ldots p_i$에 key stream $K = k_1\, k_2 \ldots k_i$를 각각 적용하여 암호화한다.

plaintext : $P = p_1\, p_2\, p_3 \ldots p_i$

ciphertext :

$$C = E_{k_1}(p_1)\, E_{k_2}(p_2) \ldots E_{k_i}(p_i)$$

암호문 $c_i = E_{k_i}(p_i)$에서, stream cipher는 평문 문자열(plaintext string)에서 암호문 문자열(ciphertext string) $c_1c_2c_3\cdots$을 생산한다. key stream k_i는 평문 p_i의 암호화 $E_{k_i}(p_i)$에 사용된다. 따라서 평문 암호화는

$$C = k_1\, p_1 k_2\, p_2 k_3\, p_3 \ldots k_i p_i \quad \text{----------(4)}$$

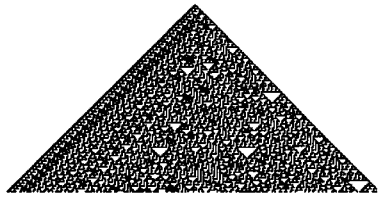
순서로 계산된다. 또한, 암호문 $c_1\, c_2\, c_3\cdots$의 복호화 과정은

$$P = k_1\, c_1 k_2\, c_2 k_3\, c_3 \ldots k_i c_i \quad \text{----------(5)}$$

순서로 계산된다. 만약 k_i가 c_i로서 해독은 $p_i = D_{k_i}(c_i)$으로 표시한다.

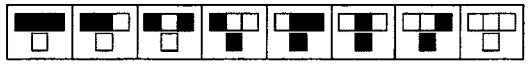
4. Cellular automata를 기초한 stream cipher

본 논문의 암호화를 위해 key 생성은 cellular automata를 이용할 것이다.



[그림 2] transition rule 30을 적용한 디자인
[Fig 2] Design that apply transition rule 30

본 논문은 stream cipher를 위해 cellular automata transition rule 30에 keystream generator 사용을 제안하였다. rule 30은 암호화 작성을 위한 key 생성에 알맞은 순환 구조를 가지고 있다. n 셀들의 1차원 배열로서, 각 셀은 2개 상태 중 하나로, 0 또는 1이 있다.



[그림 3] rule 30의 3-이웃된 1차원 배열
$(00011110)_2 = rule\,30$
[Fig. 3] One-Dimensional arrangement of 3-Neighborhood of rule 30
$(00011110)_2 = rule\,30$

셀은 3개의 이웃으로 왼쪽, 오른쪽, 그리고 그 자신이 셀이다. 입력들이 3개의 이웃 셀들의 상태들이고, output이 셀의 다음 state인 next-state function이 있다. CA(Cellular Automata)는 각 단계의 순서를 통하여 발전하며, 각 셀의 다음 state는 계산된다, 모든 셀은 동시에 그들의 다음 states로 변화한다.

$a = Current\ state\ of\ cell((i-1) \bmod n)$
$b = Current\ state\ of\ cell(i)$
$c = Current\ state\ of\ cell((i+1) \bmod n)$

$b' = Next\ state\ of\ cell(i)$

정리하면,

$b' = a \oplus (b \vee c)$ 또는
$q_i' = q_{i-1}(t) \oplus (q_i(t) + q_{i+1}(t))$ ----- Rule 30

cellular automata에 각 q_i는 0 또는 1의 값을 가지면서, N 셀들과 순환 레지스터로 이루어져 있다. 여기서 transition rule 값에 의하면 독립된 장치에 대한 시간 단계(discrete time steps)가 동시에 업데이트 된다. 본 논문에서는 순환 레지스터에 1차원 cellular automata의 transition rule 30에 의하여 생성된 key stream을 사용하는 평문에 XOR 암호화를 수행한다. 여기에 사용된 레지스터는 50 비트 길이이다. 그러나 이것은 global variable "n"의 값 변화하는 것에 의하여 변화될 수 있다. automaton register의 초기상태는 사용자에 의하여 공급된 암호에 의하여 정해진다. register의 초기상태는 seed 또는 CA에 의해 생성된 key를 사용한다. 시간 t를 통하여 특별한 셀에 의하여 획득된$q_i^{(t)}$ 값은 임의 순서(random sequence)로서 진행된다. 암호문 c는 보통 $C_i = P_i \oplus q_i^{(i)}$와 마찬가지로 binary plaintext p 로부터 얻어진다. 암호문 c에서 평문 p는 $q_i^{(t)}$가 연속이 알려져 있을 때만 같은 조작을 반복하는 것에 의하여 되찾아질 수 있다.

```
encrypt(String key,String plain) {
  plain = getStrToBin(plain);
  int len = ((plain.length()*2)+n);
  key = keyStream(key,len);
   //생성된 KeyStream 적용
  String cipher = "";
  int outBitC = 0;
  for(int i=0;i<plain.length();i++) {
     outBitC = ((int) plain.charAt(i))
                  ^ ((int) key.charAt(i));
```

```
        // C_i = P_i ⊕ q_i^(i)
    cipher += Integer.toString(outBitC);
                    }
    cipher = getBinToHexStr(cipher);
// cipher를 2진에서 16진으로 변환
  } // ends encrypt()
```

4.1 XOR encryption

암호문(ciphertext) 생성을 위하여 XOR 연산을 이용하여 피연산자들로서 평문과 cellular automata로 생성된 key stream로부터 계속된 문자들을 이용한다. XOR encryption의 작업은 평문(plaintext)으로부터 문자 즉 단일코드 값들 즉 8비트 2진수 $2^8 = 256(0\sim255)$로 나타낼 수 있다. 또한 암호문 작성은 XOR 연산에 비연산자의 평문과 연속 비트를 가지고 있는 key stream에 의해 제작된다.

비트 수	1	2	3	4	5	6	7	8	결 과
평문 (plaintext) "c"	0	1	1	0	0	0	1	1	$c(63)_{16}$
암호화 적용 keystream "m"	0	0	1	0	0	1	0	1	$m(25)_{16}$
암호문 (Ciphertext) XOR	0	1	0	0	0	1	1	0	$(46)_{16}$

<표 1> Ciphertext = Plaintext XOR Keystream

<table. 1> Ciphertext = Plaintext XOR Keystream

즉 평문의 첫 번째 문자가 "c"이면, 2진수는 01100011(16진수는 63)이다. 그리고 keystream의 첫 번째 문자가 "m"이면, 2진수는 00100101(16진수는 25)이면, XOR 연산을 적용하여 표와 같은 결과를 얻는다. 암호화 key "m"을 적용하여 평문 "c"를 XOR 연산자를 사용하여 암호화한 결과 "01000110"(16진수 46)이다. 이 과정은 각 비트별 XOR 연산자를 이용하여 암호문을 작성하며, 전체 평문이 암호화되었을 때까지 반복한다. 다음 문장에서, plaintext "p", keystream "k" 그리고 ciphertext "c"의 관계를 다음과 같이 비교해 볼 수 있다.

① p ⊕ k = c;
② 또는 c ⊕ k = p;
③ 그외 (p ⊕ k) ⊕ k = p;

XOR 연산자를 이용한 암호화 및 해독 사이클은 모든 조건에 만족한다. 따라서 bit stream cipher로서 XOR 연산자는 symmetric key encryption에 적합하다.

5. 암호화 실험

data encryption을 위해 keystream 생성은 cellular automata를 이용한다. n 셀들의 1차원 배열로서, 각 셀은 2개 상태 중 하나로, 0 또는 1이 있다. automaton의 레지스터의 초기상태는 사용자에 의하여 공급된 암호에 의하여 정해진다. register의 초기상태는 cellular automata에 의해 생성된 key를 seed로 사용한다. 레지스터를 seed하기 위해 사용된 password는 길이와 나머지(modulo) n을 이용한다.

The Plaintext :

Give me some more time to think.

다음 key를 가지고 Cellular Automata를 이용하며 메시지는 암호화된다.

keystream : 10010100

The Ciphertext :

cfd1a096c019f84e29fcdb4a088b7c7c67872b2d829b346ae5690008f05389a4

결과적으로 cellular automata를 이용하여 2진 암호화 처리를 위해 key를 생성하였으며, XOR 연산자를 이용하여 2진 암호화를 수행하였으며, 암호문의 길이를 줄이기 위해 16진법으로 작성하였다.

6. 결론

본 논문은 1차원 cellular automata transition rule 30에 keystream generator 사용을 제안하였다. 특히 cellular automata는 transition rule 30에 의하여 복잡한 형태의 암호화를 처리할 수 있었다. 앞에서 언급된 대로, cellular automata는 state, neighborhood, transition rule 이라는 단순한 성질을 가지고 다양하고 복잡한 암호화 원리를 구현할 수 있다. 따라서 본 논문에서는 이러한 cellular automata rule을 이용하여 2진 암호화 처리를 위해 key를 생성하였으며, XOR 연산자를 이용하여 2진 암호화를 수행하였다. 그 결과 기타 덧셈이나 뺄셈에 의한 암호화 즉, vigenere cipher, 문자 이동(shift cipher) 및 전치(permutation cipher), 치환(substitution cipher)에 의해 암호화하는 것, 즉 비대칭 키 암호화보다 단순히 XOR 연산을 수행함으로서 빠른 수행 속도로, 즉 symmetric key encryption로 한 단계 나은 암호화 기법으로 사료된다.

참고문헌

[1] Guan, P., "Cellular Automaton Public-Key Cryptosystem," Complex Systems 1, pp. 51-56, 1987.

[2] http://atlas.wolfram.com/01/01/.

[3] Menezes, A., van Oorschot, P., Vanstone, S. Handbook of Applied Cryptography, CRC Press, 1997.

[4] Tomassini, M., Perrenoud, M., "Stream Ciphers with One and Two Dimensional Cellular Automata," in Proceedings of the Parallel Problem Solving from Nature PPSN VI, Springer-Verlag LNCS 1917, pp. 722-731, 2000.

남 태 희

1993~현재: 동주대학 의료기공학과 부교수
1996~ 부경대학교 전자공학과 박사수료

관심분야 : Cryptography, Cellular Automata, 데이터베이스, 전자상거래, 패턴인식, MIS, GIS, ERP, 보건의료정보학