RRM기반 키 관리 방안에 의한 전자문서 암호화에 관한 연구

성경상, 오해석
경원대학교 전자계산학과
e-mail:actofgod@ku.kyungwon.ac.kr
oh@kyungwon.ac.kr

A Study on e-Document Encryption using Key Management Method based on the RRM

Kyung-Sang Sung, Hae-Seok Oh
Dept of Computer Science, Kyungwon University

요 약

전자문서를 대상으로 하는 다양한 보안 기술들이 연구·제시되고 있으나, 키 관리에 대한 어려움과 암호 알고리즘의 무거운 특성으로 안전성과 효율성의 반비례 관계가 발생하고 있다. 본 연구의 목적은 위와 같은 문제를 해결하기 위해 전자문서 암호 시스템에 적용 가능한 제안하는 RRM 기법을 응용하여 키 관리 방안에 적용함으로써 효율적인 암호화 과정을 수행하여 전자문서 보호 문제를 개선하는 것이다.

이를 위하여 난수정보에 규칙성을 부여함으로써 키 생성에 대한 어려움을 극복하고 키 테이블과 키셋 정보를 통해 키 관리 문제를 해결하며, 키셋 정보를 통해 복호화를 위한 연산 수행속도를 빠르게 진행할 수 있는 개선된 전자문서 암호화 시스템 수행을 위한 키 관리 방안을 제안한다.

제안하는 키 관리 방안을 통해 키 생성 연관성 문제를 해결함으로써 키 노출문제에 대한 안정성과 단순한 암복호화 과정에 비해 동일한 복잡도와 수행시간을 갖는 연산 기법을 이용하여 효율성을 높였으며, 전자문서를 암호화 수행 후 관리를 함으로써 유출문제에 대한 문제도 해결할 수 있다.

키워드 : 전자문서, RRM(난수 재배열 방법), 키 관리, 난수정보, 암복호화, 연산 기법

I. 서론

인터넷 사용의 보편화와 정보통신 기술의 발달은 종이 문서에 의한 기업의 환경을 급격히 변화시키고 있으며 전자적인 방법으로의 대체를 통해 새로운 형태의 사업 기회를 제공하고 있다. 이와 같은 전자문서의 등장은 종이문서의 작성, 보관에 따른 고비용 구조를 큰 폭으로 개선시켰으며, 정보처리에 소요되는 물자와 노력이 절약되어 거래 비용을 절감하며 시장의 투명성을 증대시킴으로써 조직의 생산성과 효율성을 증대시키고 궁극적으로는 국가경쟁력을 제고시킨다.

이와 같이 전자문서를 이용함으로써 많은 이점을 가지게 되지만 동시에 보안상의 많은 위협을 가지게 된다. 즉, 전자문서의 무단 유출, 파괴, 분실·훼손의 위험이 항상 존재하고 있을 뿐만 아니라 관리에 따른 어려움도 뒤따른다.

위와 같은 문제를 해결하기 위해 문서의 무결성 및 보안성을 확보하기 위한 암호 기술 및 불법복제 방지 기술, 권한 기반 접근, 열람 및 편집을 지원하기 위한 접근제어 기능, 편집 기록을 관리하기 위한 시점확인 기술, 증명 기술 등이 필요하다. 전자문서를 대상으로 하는 다양한 보안 기술들이 연구·제시되고 있으나, 대부분 위·변조 및 부인방지에만 국한되어 있다. 전자문서의 암호화를 모색함에 암호 방식의 무거움으로 인해 전자문서에 대한 직접적인 암호화는 추진하고 있지 않고 있다. 또한 암복호화 키 관리의 어려움을 이유로 보안에 대해서는 소홀히 대처하고 있으며, 안전성과 효율성의 반비례 관계가 발생하고 있다.

이러한 관점에서 전자문서 암호화 기술을 이용한 전자문서 관리 시스템을 설계하고 안전한 전자문서를 제공하기 위해, 전자서명 기술을 통한 위변조 및 부인에 대한 문제를 해결하고, 전자문서 암호화 시스템에 적용 가능한 RRM을 이용한 효율적인 키 관리 방안에 관한 연구를 통해 전자문서 자체의 유출 문제를 개선할 수 있도록 한다.

RRM이란 Random Rearrangement Method의 약자로서, 랜덤하게 발생되는 난수 정보를 재배열함으로써 규칙성을 부여하도록 하였으며, 규칙성이 부여된 난수 정보를 이용하여 문제가 되고 있는 암복호화 키 관리 문제를 해결하기 위해 본 논문에서 제안하는 방법이다.

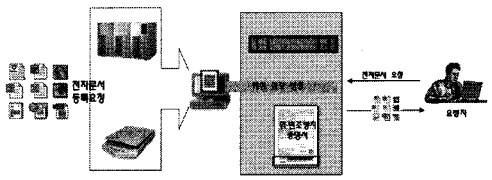
제안하는 방법을 통한 암복호화 과정은 단순하게 이루어지지만, 동일한 복잡도와 수행시간을 갖는 XOR 연산 기법을 이용하여 효율성을 높였으며, 대칭키 암호 표준인 AES 암호화 알고리즘과 난수 재배열에 규칙성을 부여하여 생성된 키 관리 방안을 이용하여 전자문서를 보다 안전하고 효율성을 개선한 시스템 구축을 통해 기존 전자문서 암호화 시스템에서 운용하는 암호화 시스템과 성능 비교 평가하였으며, 안정성과 효율성 모두 개선된 결과를 얻을 수 있었다.

II. 관련 연구

본 장에서는 전자문서 시스템의 문제점을 분석하기 위해 전자문서 위변조를 적용한 시스템과 전자문서에 암호 알고리즘을 적용한 시스템에 대한 문제점을 파악하고, 개선안을 제시한다.

2.1 전자문서 위·변조를 적용한 시스템의 개선점

전자문서의 위·변조만를 검증하는 시스템 흐름은 〈그림 1〉과 같은 과정을 지닌다.



〈그림 1〉 위·변조만을 검증하는 전자문서 시스템

〈그림 1〉에서 보는 바와 같이 전자문서는 이미지 정보의 특징을 지니고 있으며, 헤더부에는 이미지의 위변조를 확인하기 위해 이미지 정보를 SHA1 알고리즘을 이용하여 해쉬화한다. 전자문서 관리 서버는 해쉬된 이미지 정보를 헤더부에 집어넣고 관리하며, 이후 요청자 또는 해커에 의해 해당 전자문서의 전반적인 내용이 노출되는 문제가 발생된다. 위와 같은 방식의 전자문서를 관리 방안은 해당 전자문서의 위·변조 여부만을 가리기 위한 것이므로 그 내용에 대한 유출여부 문제는 책임 여부를 확신할 수 없다.

또한, 전자문서는 일반 텍스트 문서의 특성이 아닌 전자화 정보를 지닌 이미지 특성을 가진다. 단일 페이지 문서라면 암복호화하는데 큰 무리가 따르지 않겠지만, 멀티 페이지를 가진 전자문서라면 이미지 특성상 그 용량은 무시할 수 없게 된다. 이렇게 큰 문서를 암호화해서 요청자에게 전송하고, 송신자는 전달받은 암호화된 전자문서를 복호화하는데 많은 시간적 비용과 시스템적 트래픽에 대해 보장받을 수 없게 되는 문제를 지니게 된다.

이와 같은 문제를 해결하기 위한 방안으로 빠른 연산 속도를 지닌 대칭키 기반 암호 알고리즘을 이용하여 암호화 시스템에 적용하고 있으나, 키 관리에 대한 문제점을 지닌다.

2.2 기존 전자문서 암호화 시스템의 개선점

전자문서를 암호화하기 위한 방안으로 정보의 접근권한 레벨에 따라 암호화하는 방법과 중요도에 따라 여러번 암호화하는 방법이 이용되고 있으며, 해쉬함수를 이용한 키 생성을 통해 전자문서의 연관된 관계를 응용한 일방향 키 체인 방식과 전자문서에 대한 등급별 보안을 적용한 방식과 선택적 암호화에 관한 방식들이 소개되고 있다.

2.2.1 일방향 키 체인 방식 시스템

일방향 키 체인 방식은 역함수가 존재하지 않는다는 해쉬함수의 특징을 이용하여 키를 생성하고 관리하는 방법으로, 기존에 존재하는 비공개키 암호화 방식에서의 키 관리에 대한 어려움을 극복하기 위해 제안된 방식이다. 이러한 방식은 상위 레벨키를 가진 사용자는 하위 레벨키들을 도출해 낼 수 있지만, 그 역은 불가능하다는 뜻이다.

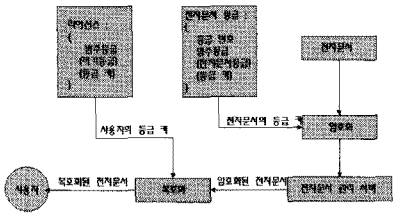
이러한 방법론은 손쉽게 키 관리를 할 수 있는 장점은 가지고 있지만, 키 우선순위에 대한 개념 정립에 문제가 발생됨으로써 접근제어에 관한 권한을 부여받은 사용자의 무분별한 행위에 대해서는 보장받을 수 없다. 또한, 암호화된 전자문서의 중간에 접근하기 위해서는 차례대로 키를 복호화해야 하며, 초기 키값이 유출되거나 손상된 경우에는 해당 전자문서에 대해서는 보장받을 수 없다는 문제점을 지닌다.

2.2.2 등급별 보안을 적용한 암호화 시스템

전자문서 서비스를 제공하는 업체는 사용자와 해당 전자문서를 어떠한 기준에 의하여 여러 그룹으로 나누고, 각각에 대하여 접근 제한 정책을 사용한다. 이와 같이 사용자와 전자문서에 등급을 부여하여 접근을 제한하는 것을 등급별 보안이라 한다.

전자문서의 암호화 과정은 등급 정책에 부합하는 등급키를 랜덤으로 생성하고, 〈그림 2〉와 같이 전자문서 등급에 부합하는 등급키를 이용하여 암호화한다. 복호화는 사용자가 자신이 소유한 라이선스에서 복호화에 필요한 키를 추출하여 수행한다. 일반적으로 전자문서의 암호화와 복호화에 사용하는 Rijndael 암호 알고리즘을 이용하며 128 비트 키를 생성

하여 사용한다. 암호화 과정을 거친 전자문서는 자신의 라이선스에서 등급키를 추출한 후 복호화 과정을 수행한 후 관련 전자문서에 접근하는 과정을 거친다.



〈그림 2〉 전자문서의 암호화와 복호화

등급과 범주에 따라서 접근 제한을 두는 방법은 전자문서를 관리하는 입장에 있어서는 매우 편리하게 접근할 수 있다. 사용자의 권한보다 높은 등급의 전자문서에 접근할 수 없게 하고, 해당 등급 이하의 전자문서는 모두 접근하도록 하기 위하여 등급에 따라서 여러 개의 키를 사용하는 방법은 매우 위험할 수도 있다. 즉, 등급에 따라 구분되어지는 접근 권한 여부에 따라 자신 밑에 존재하는 모든 문서에 대해서는 슈퍼 관리자와 같은 권한이 부여되기 때문이다.

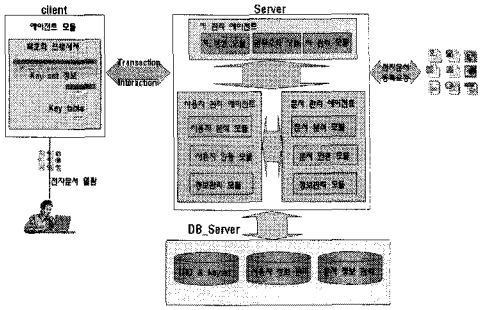
또한, 개별 인증이 아니라 등급별 인증 방법을 사용하고, 전자문서 배포시 사용자 인증절차를 요구하지 않으므로 해당 전자문서나 사용자 등급 조정에 따라 발생되는 문제점에 대해서는 전혀 고려되지 못하고 있다.

III. 제안하는 시스템

서버는 키 관리 에이전트와 사용자 관리 에이전트, 문서관리 에이전트 그리고 데이터베이스로 구성되며 클라이언트는 암호화된 전자문서를 열람하기 위한 복호화 모듈로 구성되어 있다.

1.1. 3.1 제안 시스템의 전체 구조

1.2.



〈그림 3〉 제안하는 시스템 구조

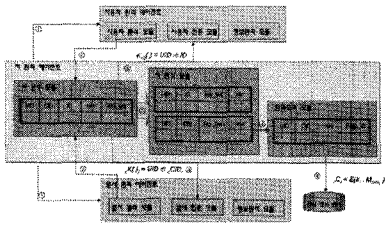
먼저 전자문서의 등록 요청이 발생되면 정해진 규정에 따른 전자화 관리 시스템을 통해 전자화 과정을 거치게 된다. 검증 시스템을 통해 전자화 과정을 거치게 된다. 전자화 과정 속에서 문서가 갖는 속성 정보로서 문서의 작성자, 제목, 작성일, 등록일, 주제어, 설명, 보안등급 등 문서를 등록할 때 작성하는 메타데이터가 기록되며, 위와 같은 정보들은 해당 절차에 따라 관리된다. 등록된 전자문서의 분석된 정보를 기반으로 키 관리 에이전트부의 RRM 기법을 이용하여 키 생성 모듈을 통해 문서의 암호화를 수행하기 위한 키를 생성한다. 생성된 키를 이용하여 암·복호화 모듈에서는 전자문서의 개별 페이지 정보에 대한 암호화를 대칭키 기반의 AES 알고리즘을 이용하여 수행한다. 암호화를 수행한 키의 노출을 막기 위한 방안으로 개별 관리 방법을 이용한다. 전자문서 열람을 원하는 사용자는 정당한 인증과정을 거쳐 등록되며, 사용자 관리 에이전트부의 사용자 정보 모듈을 통해 등록된 사용자의 정보는 분석된다. 분석된 사용자 정보는 이름, 아이디, 주민등록번호, 직업 등의 메타데이터로 정리되며, 위와 같은 정보들은 해당 절차에 따라 관리된다. 사용자를 위한 s-Box 역할을 수행하는 키셋 정보 생성을 위해 사용자 인증 모듈에서는 키 관리 에이전트를 호출한다. 분석된 사용자 정보를 기반으로 RRM 기법을 이용하여 키 생성 모듈을 통해 64바이트의 고유한 키셋 정보를 생성하고 관리하는 시스템 구조를 지닌다.

3.2 키 관리 에이전트

등록된 사용자와 전자문서를 암호화 수행을 위한 키 관리 에이전트는 키 생성 모듈과 암·복호화 모듈 그리고 키 관리 모듈로 구성된다.

키 생성 모듈은 UID, CID, ID 정보로 구성되며, UID는 전자문서에 부여된 고유 정보를 나타내며, CID는 등록된 전자문서에 사용되는 아이디 정보를 지니고, ID는 사용자가 등록할 때 사용한 아이디 정보를 가진다. 그리고 키 필드는 이와 같은 정보들을 이용하여 생성된 암호키를 관리하고, 키셋 필드는 사용자의 키셋 정보를 관리하기 위한 곳이다.

각 모듈의 구성은 〈그림 4〉와 같으며, 상세한 처리 과정은 다음과 같다.



〈그림 4〉 키 관리 에이전트 처리절차

① 키 관리 에이전트는 키를 생성하기 위해 사용자 관리 에이전트와 문서 관리 에이전트에게 관련 정보를 요청한다.
② 각 에이전트부의 분석 모듈을 통해 키 생성에 필요한 정보를 키 생성 모듈에게 전송한다.

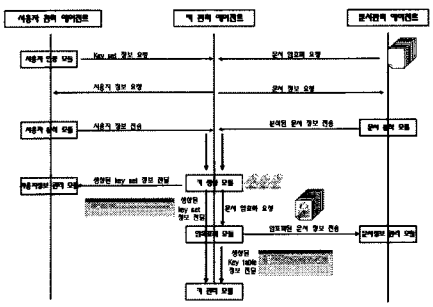
③ 키 생성 모듈은 사용자의 UID 정보와 ID 정보를 이용하여 키셋 정보를 생성($K_{ID}\{\} = UID \oplus ID$)한 후 사용자 인증 모듈과 키 관리 모듈에게 전달하고, 문서의 UID정보와 CID정보를 이용하여 개별 키를 생성($_nK\{\}_i = UID \oplus {}_nCID_i$)한 후, 문서 인증 모듈과 키 관리 모듈에 전송한다.

④ 생성된 키 정보를 이용하여 암·복호화 모듈을 통해 전자문서를 개별 페이지별로 암호화 과정($_nC_i = E(k_i, M_{num\_i})$)을 수행한 후 암호화된 전자문서는 문서 정보 관리 DB에 저장 관리된다.

3.3 키 관리 프로토콜 설계

본 논문에서 제안하는 시스템은 키 관리 에이전트를 기준으로 구성되며, 생성된 키는 대칭키 기반 암호화 수행을 위한 암호키로 활용한다. 키 생성을 위해 무작위로 난수를 발생하고 규칙성을 부여하기 위해 비트 패턴을 이용한 XOR 기법을 활용하였다. XOR 기법은 연산 수행 속도가 빠른 특징을 가지며, 암·복호화 과정을 단순하게 수행하지만, 송·수신측에서는 동일한 복잡도와 수행시간을 갖는다.

키 관리 프로토콜의 전체 구조는 〈그림 5〉와 같다.



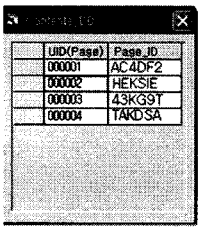
〈그림 5〉 키 관리 프로토콜

사용자 관리를 위한 정보를 구하기 위해 사용자 정보관리 모듈에서는 키 관리 에이전트에게 고유키(키셋 정보)를 요청한다. 키 관리 에이전트는 키셋 정보 생성을 위해 필요한 사용자 정보를 요청하며, 사용자 분석 모듈을 통해 해당 정보를 전송한다. 사용자 정보를 기반으로 키 생성 모듈을 통해 생성된 키셋 정보는 키 관리 에이전트에 보관하고, 사용자 인증 모듈에게 해당 정보를 전송한다.

문서관리 에이전트로부터 문서 암호화 요청이 발생된 경우, 키 관리 에이전트는 암호키 생성을 위해 분석된 문서 정보를 문서 관리 에이전트부의 문서 분석 모듈을 통해 요청한다. 분석된 문서 정보를 기반으로 키 생성 모듈을 통해 암호화 수행을 위한 키를 생성한다. 암·복호화 모듈을 통해 생성된 키를 이용하여 요청된 전자문서를 암호화를 수행한 후, 키 정보는 테이블로 구성하여 키 관리 모듈에서 보관한다.

3.4 암호키 생성을 위한 설계

전자문서를 암호화하기 위한 구성요소로는 페이지를 구분할 수 있는 정보와 각 페이지의 암호화 수행을 위한 키 매칭 정보를 가져야 한다. 제안하는 시스템에서는 이와 같은 조건을 만족하기 위해 〈그림 6〉과 같은 요소로 구성하였다. UID 정보는 전자화 과정을 거쳐 분석된 페이지 넘버를 나타내며, page_ID는 각 페이지와 매칭되는 암호키 정보를 가리킨다. 이와 같이 전자문서에 대해 각 페이지별로 암호화를 수행한 키 정보를 테이블 형식으로 매칭시켜 관리하는 것을 목적으로 한다.



UID(Page)	Page_ID
000001	AC4DF2
000002	HEKSIE
000003	43KG9T
000004	TAKDSA

〈그림 6〉 문서관리를 위한 키

정의된 64개의 문자를 이용하여 불규칙적인 난수 정보에 규칙성을 부여하여 문자 6개로 구성된 키 값을 생성한다. 본 논문에서 제안하는 키 길이는 가변적 성격을 띄고 있으며, 키 길이에 따라 보안 강도는 달라진다. 문서의 중요도에 따라 키 길이가 늘어나도 XOR하는 비트 패턴을 통해 암·복호화 과정이 단순하게 이루어지지만, 동일한 복잡도와 수행시간을 가지므로 제안하는 키 관리 방식의 효율성은 뛰어나다고 할 수 있다.

〈그림 7〉은 문서에 사용할 암호키 생성을 위해 의사코드화로 표현한 것이다.

```
procedure Array Doc_key(N)
{
  if Doc_code != Enroll
    call Doc_code;
    call key_set_table;
    for i := 1 to 6
      if (i == 1)
          temp = CID XOR A(i)
      else
          temp = UID XOR A(i)
    next
    Doc_key_set(N) = temp XOR Doc_code
    return Doc_key_set(N)
}
```

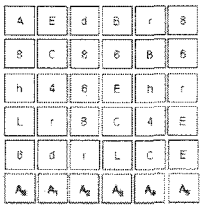
〈그림 7〉 암호키 생성 의사코드

키 생성을 위해 등록된 문서인지 확인한 후, 등록되어 있지 않으면 Doc_code를 호출한다. 먼저 6개의 null을 생성한 후, 정의된 키셋 테이블을 통해 첫 번째 값부터 A(6)번째 열까지 채워나가는데, 첫 번째 값은 분석된 문서에 등록된 CID

정보와의 연산 수행을 통해 획득하게 되며, 생성된 이전 값과 UID 정보를 XOR 연산 수행함으로써 새로운 값을 얻게 된다. 이와 같은 과정을 반복해 얻은 값은 문서를 암호화하기 위한 키로서 대칭키 역할을 수행한다.

3.5 키 배열을 통한 복호화 과정

본 논문에서는 전자문서 관리를 위해 불규칙적으로 배열된 난수 정보에 규칙성을 부여하여 생성된 키를 이용하여 XOR 연산을 통해 암호화를 수행한다. 이와같이 암호화된 전자문서를 복호화하기 위한 방안으로 키 배열에 따른 규칙성을 파악한 후, 사용자의 키셋 정보를 기반으로 복호키를 추출해 내는 과정을 수행한다.



〈그림 8〉 패딩(padding)된 키 테이블

복호키 생성을 위해 키셋 정보와의 XOR 연산 수행을 통해 패딩된 키 테이블 정보로서, 초기값은 null값으로 채운 후, 마지막 행을 제외한 1~5행은 정의된 사용자 키셋 정보를 기반으로 연산수행을 통해 생성된 데이터를 순서대로 채워나가고 마지막행에는 복호키를 역연산하여 수행하여 산출된 결과를 패딩한 키 테이블을 생성한다.

〈그림 8〉은 마지막 행을 제외한 나머지 행에 대해 사용자의 키셋 정보를 이용하여 패딩한 결과이며, 이와같이 패딩된 값을 전자문서의 암호키값과 수식 1과 같이 연산 수행한 결과값을 마지막 행에 패딩함으로써 키 테이블을 완성한다.

$$A \oplus 8 \oplus h \oplus L \oplus 6 \oplus \text{암호키값} = A_0 \qquad \text{수식 1}$$

```
procedure Array key_table(N)
    call user_key_set;
{
    for i := 1 to 5
      if (i == 1)
          temp = user_key_set XOR A(i)
    else
          temp = temp XOR A(i)
    next
    key_table(N) = temp + key-1(N)
    return key_table(N);
}
```

〈그림 9〉 키 테이블 생성 의사코드

완성된 키 테이블과 암호화된 전자문서를 함께 사용자에게 전송한다. 〈그림 9〉는 수식 1을 기반으로 키 테이블을 생성하기 위해 의사코드를 통해 표현한 것이다. 의사코드화된 〈그림 9〉를 통해보면, 키값을 제외한 1~5행은 호출된 사용자 키셋 정보를 기반으로 연산수행을 통해 키 데이터를 생성하고, 마지막 행은 암호키값과의 역연산 수행을 통해 키 테이블이 생성된다.

IV. 성능평가 및 분석

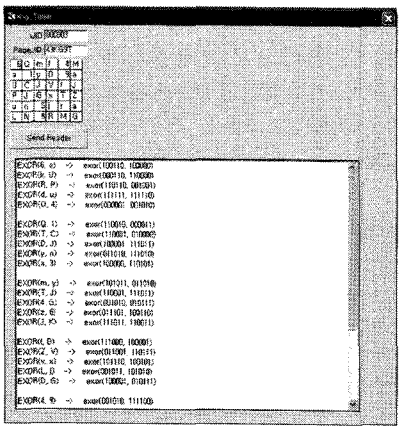
본 장에서는 먼저 구현 환경 및 실험 개요를 기술하고, 암·복호화의 안전성을 분석하여 성능평가를 수행한다.

1.3. 4.1 구현 환경

제안하는 시스템 개발을 위한 환경은 시스템 Intel (R) Pentium (R)-4 CPU 2. 66GHz와 2GB RAM, 그리고 MS-WindowS XP Professional 운영체제를 사용 하였다. 또한 구현 언어는 Visual Basic 6.0을 이용하여 키 정보에 대한 전반적인 내용과 서버 및 클라이언트 인터페이스를 구현하였다. 전자문서의 UID 값과 암호키, 키셋에 대한 정보를 저장하기 위한 데이터베이스는 MS- SQL 2000 프로그램을 사용하였다.

1.4. 4.2 실험 개요

〈그림 10〉은 키 테이블을 생성할 때 정당성을 보이기 위한 검증 과정을 함께 나열한 것이다. 현재 보여지는 UID 3번의 Page_ID 값은 43KG9T로서 패스워드 정보를 대체한다. 패스워드 정보를 유추하는 과정은 키 테이블의 첫 번째 열 부분을 통해 계산해 간다. 즉, 6과 a를 XOR 과정을 수행하면 k가 나오며, k와 U를 XOR 연산 수행하면 R이 나온다. 이와 같은 방식으로 계속 수행하고 마지막에 키 테이블에 위치한 L값은 도출된 O와 패스워드 4를 XOR 연산 수행한 결과가 된다.



〈그림 10〉 키 테이블 검증 과정

그러므로 사용자 Alice에게 테이블 정보를 전송하게 되면 사용자 Alice는 자신의 키셋 정보를 기반으로 하여 키 테이블을 XOR 연산 수행하면 정확한 Page_ID 즉 패스워드 값을 얻게 된다.

1.5. 4.3 실험 및 평가

가장 많은 시간을 소요하는 것은 대용량 콘텐츠에 대한 암복호화 시간이며, 안전성에 대해서도 평가를 하고자 한다. 본 논문에서 평가를 위해 전자문서 시스템에서 이용하는 암호방식을 비교대상으로 하며, 제안하는 암호키 생성 방식을 전자문서 시스템에 적용한 것을 두고 분석하였다.

4.3.1 암호화에 대한 비교 분석

암호화에 대한 비교분석은 〈표 1〉과 같이 암호화 기법과 키의 노출 가능성, 전자문서 자체의 암호화 방식과 적용파일에 대하여 비교해볼 수 있다.

〈표 1〉 전자문서 시스템과의 암호화에 대한 분석

비교 항목	기존의 RSA 기반 시스템	제안하는 개선된 전자문서 시스템
암호화 기법	공개키 방법 적용	복수개의 대칭키
키노출가능성	낮음	낮음
키 관리에 대한 어려움	키 관리에 따른 편리함 키 노출로 인해 서버와 클라이언트측에서 재 관리 방식 적용	클라이언트 : 키 셋 정보만 관리함으로 편리 서버 : 키 테이블 형식을 가지므로 관리의 편리 - 키 노출에도 암호화된 문서 유출 우려 없음
암호적용방식	전자문서의 전체 내용	전자문서의 개별 페이지
암복호화 속도	문서 크기에 따라 다름 상대적으로 오래 걸림	일정한 속도를 유지 빠른 연산 수행속도를 유지함
암호화 강도	강함	강함
보관 방식	평문	암호문

1.6.

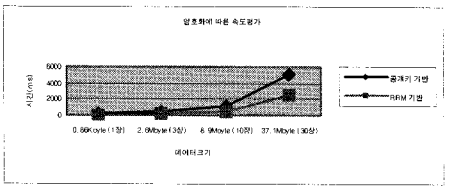
4.3.2 속도적 성능 평가

본 논문에서 실험 평가를 위해 TIFF 형식의 전자문서를 데이터로 사용하였다. 실험 평가를 위해 3장 분량의 2.6Mbyte 크기의 전자문서와 10장 분량의 8.9Mbyte 크기의 전자문서, 30장 분량의 37.1Mbyte 크기의 전자문서를 데이터로 사용하였다.

〈표 2〉 기존 시스템과의 암호화에 대한 시간 (ms)

데이터 크기 (전자문서)	공개키 기반	RRM 기반
0.86Kbyte (1장)	140	78
2.6Mbyte (3장)	500	234
8.9Mbyte (10장)	1234	421
37.1Mbyte (30장)	5048	2512

암호화에 대한 시간을 비교 분석한 결과는 제안한 방식을 통해 암호화를 수행한 결과는 〈표 2〉와 같이 기존의 전자문서 시스템 보다 향상되었음을 보이며 〈그림 11〉과 같이 보여진다.



〈그림 11〉 암호화에 따른 속도 평가

V. 결론

제안하는 전자문서의 개별 정보들을 암호화하는 방식을 통해 전자문서를 암복호화하는데 발생되는 시간과 시스템적 비용을 개선할 수 있었으며, 요청한 개별 정보만을 수신함으로써 불필요한 정보에 대해 처리해야 하는 문제와 더불어 다른 관련된 중요한 문서의 노출 문제를 해결할 수 있었다.

RRM을 이용한 효율적인 키 관리 방안을 통한 전자문서 암호화 시스템을 구축함으로써 시스템의 유연성을 책임질 수 있게 되었다. 또한 생성한 키를 이용하여 전자문서의 개별 관리를 함으로써 문서 노출에 대한 문제를 해결할 수 있었으며, 복호화에 필요한 키 테이블 정보가 전송 중 손실되어도 키 테이블의 재전송 요청을 통해 해결할 수 있다. 키 테이블 정보는 연산 수행할때마다 다른 결과를 보이므로 안전한 보안 서비스를 제공할 수 있었다.

제안하는 시스템은 기존 해쉬 알고리즘을 이용하여 위변조 문제를 해결한 후 RRM을 이용한 키 관리 방안을 적용한 암호 알고리즘을 이용하여 전자문서에 대한 암호화를 할 수 있으며, 향후 전자문서에 대한 보안적 측면이 확산되었을 때, 제안 기법의 알고리즘과 구현 방법은 다양한 응용분야에서 기본 모델로 활용이 가능하다.

참고문헌

[1] 한국전자거래진흥원,"전자화문서의 생성 방법 및 절차에 관한 지침 소개", 2006. 8

[2] 한국정보보호진흥원,"암호이용가이드라인", 2007.

[3] Australian Government e-Authenticaiton Framework Implementation Guide for Government, AGIMO, 2005. 3

[4] R.OWENS, R.AKALU, "Legal Policy and Digital Rights Management," IEEE Proceedings, VOL. 92 NO. 06 pp. 997 ~ 1003 2004. 06