情 報 保 護 學 會 誌
第20卷 第5號, 2010. 10

스마트그리드 환경에 적용 가능한 암호화된 데이터베이스 검색 기술 동향 분석

김 기 성*, 김 동 민*, 노 건 태*, 정 익 래*

요 약

아날로그 기반의 전력 시스템과 디지털 기반의 IT 기술을 융합한 스마트그리드는 최근 가장 주목받는 분야 중의 하나이며, 이것은 기존의 전력 생산, 소비, 운반 프로세스에 IT 기술을 접목하여 전력 효율의 극대화를 추구한다는 장점이 있다. 하지만 스마트그리드 환경에서는 기존의 아날로그 기반의 전력 시스템에서 문제가 되지 않았던 부분이 디지털 기반의 IT 기술과 융합되면서 새로운 보안 위협들로 발생할 우려가 매우 높다. 따라서 본 고에서는 스마트그리드 환경의 보안 위협을 분석하고, 이러한 보안 위협을 해결할 수 있는 방법으로 암호화된 데이터베이스 검색 기술의 동향을 살펴본다. 암호화된 데이터베이스 검색 기술은 최근 발생한 수많은 데이터베이스 유출 사건을 계기로 활발히 연구되고 있는 분야이며, 암호화된 데이터를 복호화하지 않고도 효과적으로 검색할 수 있는 기술에 대한 연구이다.

Ⅰ. 서 론

아날로그 기반의 전력 시스템과 디지털 기반의 IT 기술을 융합한 스마트그리드는 2000년 이후 전 세계적으로 가장 주목받는 분야 중의 하나이다. 스마트그리드는 기존의 아날로그 기반의 전력 시스템에 IT 기술을 접목하여 전력 거래소와 사용자가 실시간으로 정보를 교환하며, 전력 사용 및 공급의 효율성을 극대화한다. 스마트그리드 환경에서 사용자는 전력 요금이 저렴할 때 전기를 예약하여 사용할 수 있고, 전자제품이 스스로 전력 요금이 저렴한 시간대를 선택하여 작동하도록 설정하는 것 역시 가능하며, 전력 거래소에서는 이러한 전력 사용 현황을 실시간으로 파악하기 때문에 기존의 아날로그 기반의 전력 시스템에서와 같이 전력 예측량의 10% 이상을 예비 전력으로 준비할 필요가 없어 실시간으로 전력 공급량을 탄력적으로 조정할 수 있다. 스마트그리드 환경에서는 가정, 사무실, 공장 등 어느 곳에서든지 자신이 사용한 전력 요금을 실시간으로 확인할 수 있으며, 이러한 정보를 바탕으로 하여 전력 거래소는 전력 보유량을 효율적으로 관리할 수 있다.

기존의 아날로그 기반의 전력 시스템과 비교하여 사용자 측면에서 스마트그리드의 가장 핵심적인 변화 요인은 전력 정보에 대한 양방향 통신이다. 기존의 아날로그 기반의 전력 시스템에서는 전력과 관련된 모든 정보를 전력 거래소에서 전적으로 관리하였으나, 스마트그리드 환경에서는 전력 거래소와 사용자(전자제품)가 전력과 관련된 정보들을 실시간으로 주고받게 된다. 예를 들면, 세탁기는 인터넷을 통해 전력 거래소와 통신하며, 그 결과로 얻은 정보를 바탕으로 세탁기는 자동적으로 전력 요금이 저렴한 시간대를 선택하여 세탁을 수행하게 된다. 또 다른 예로, 여름철에 과도한 에어컨의 사용으로 전력 거래소에서 전력 관리에 비상이 생긴다면, 우선적으로 그 시간대의 비용을 증가시켜 사용량을 떨어트릴 수도 있고, 최악의 경우 전력 거래소에서 에어컨의 온도를 강제로 올리는 비상조치까지도 가능하게 된다. 이와 같이 스마트그리드 환경에서는 전력 사용과 관련된 모든 기능을 전력 거래소와 사용자 간의 양방향 통신으로 이루어지게 된다. 따라서 스마트그리드 환경에서는 필요한 만큼만 전력을 생산하거나 생산되는 전력량에 맞춰서 사용할 수 있으며, 결과적으로 전력에 대한 에너지 효율을 극대화하여 지구의 온난화와 같은 이상기후를 막는 것이 가능하다.

* 고려대학교 정보경영공학전문대학원 ({gisung2137, kkomang03, oldos, irjeong}@korea.ac.kr)

본 고에서는 스마트그리드 환경에서 발생 가능한 보안 위협에 대해 살펴보고, 이러한 보안 위협을 해결할 수 있는 방안으로 지금까지 제안된 암호화된 데이터베이스 검색 기술 중에서 스마트그리드 환경에 적용 가능한 기술들에 대해 살펴본다.

Ⅱ. 보안 위협 및 대책

기존의 아날로그 기반의 전력 시스템의 일방향 통신에서 스마트그리드의 양방향 통신으로 구조가 변화하게 되면, 기존의 아날로그 기반의 전력 시스템에서 존재하는 취약성 뿐만 아니라 현재 존재하는 인터넷 상에서 발생 가능한 대부분의 보안 위협에 그대로 노출되게 된다. 즉, 전력 거래소 내부의 해킹을 통해 대규모 정전 등의 사고를 발생시킬 수 있고, 각 가정마다 설치된 스마트미터기를 통해 개인의 신상 정보가 고스란히 노출될 수 있다. 실제로 2003년 미국 북동부와 캐나다 남부 지역의 8개 주에 걸쳐서 대규모 정전 사태가 발생하여 공항, 수도 등의 기간 시설들이 마비되고 공장 운영이 중지되어 최소 40억 달러 이상의 금전적 피해가 발생하였으며[1], 스마트그리드 환경에서는 이러한 사고가 악의적인 해커에 의해 전 세계적인 위협이 될 것이다. 즉, 이러한 대규모 정전 사태가 악의적인 해커에 의해 스마트그리드 환경에서 발생하게 된다면, 네트워크에 의한 감염 확산으로 그 피해가 더욱 빠르고 광범위하게 확산될 수 있다.

또한, 각 가정마다 설치될 스마트미터기에 대해서도 다양한 보안 위협이 존재한다. 스마트미터기는 사용자들이 가지고 있는 작은 데이터베이스이며, 전력 사용에 관한 정보뿐만 아니라 요금 결제 및 전력 관리를 위해 사용자 개인의 신상에 관한 다양한 정보들을 포함하고 있다. 따라서 스마트미터기에 저장된 정보가 악의적인 공격자에 의해 노출된다면, 사용자들의 심각한 프라이버시 문제를 야기하게 된다.

스마트그리드 환경에서 발생되는 사용자의 프라이버시 문제들은 데이터베이스 보안과 직접적인 관련이 존재한다. 각 가정에 설치된 스마트미터기는 사용자의 정보를 담고 있는 작은 데이터베이스가 되고, 이러한 정보를 수집하여 지역별로 관리하는 지역 관제소에서는 그보다 규모가 조금 더 큰 데이터베이스를 형성한다. 최종적으로 전력 거래소는 이러한 정보를 모두 포괄하여 저장하고 관리하는 대형 데이터베이스를 형성하게 된다. 이러한 상황에서 데이터베이스 보안은 스마트그리드 보안의 충분조건이 될 수는 없으나 필요조건임은 분명하다.

데이터베이스 보안은 일반적으로 접근 제어 방식과 데이터 암호화 방식으로 분류한다. 접근 제어 방식은 데이터베이스 관리자가 존재하여 각 사용자마다 접근 권한을 부여하는 방식이다. 따라서 데이터베이스에 접근하는 사용자의 인증이 우선적으로 일어나며, 인증된 사용자는 허가된 권한 내에서 데이터베이스에 접근이 가능하다. 접근 제어 방식의 가장 큰 장점은 효율적으로 데이터베이스의 접근 및 관리가 가능하다는 점이다. 접근 제어 방식은 각 사용자들에게 접근 권한을 부여하고 안전한 인증 기법을 사용한다면, 특별한 과부하를 발생시키지 않는다. 하지만 이러한 특성 때문에 여러 가지 단점이 존재하게 된다. 먼저, 데이터베이스 관리자를 전적으로 신뢰할 수 있는 환경에서만 사용이 가능하다. 즉, 데이터베이스를 아웃소싱하거나 클라우드 컴퓨팅 환경에서처럼 자신의 민감한 정보를 외부에 저장하는 환경에서는 사용이 불가능하다. 또한, 접근 제어 방식은 내부자의 공격에 매우 취약하다는 단점을 가진다. 실제로 데이터베이스 보안 사고에서 가장 커다란 부분을 차지하는 것이 내부자에 의한 공격 및 실수로 인한 노출이다. 이와 같이 접근 제어 방식은 효율성 면에서는 뛰어나지만, 다양한 위협 및 제약 조건이 뒤따르게 된다.

데이터 암호화 방식은 데이터를 저장할 때 암호화된 형태로 저장하는 것을 의미한다. 암호화 기법은 데이터를 보호하는 가장 안전한 방법이며, 이러한 암호화 기법을 사용하면 자신의 민감한 정보를 외부에 저장할 수도 있고, 내부자에 의한 공격으로부터도 안전하게 된다. 하지만 기존의 암호화 기법을 그대로 사용하여 데이터를 암호화하고 이것을 그대로 데이터베이스에 저장한다면, 매우 비효율적이다. 만약 사용자가 문서들을 암호화하여 외부에 저장하였다고 가정하면, 사용자는 특정 문서를 검색하기 위해서 저장된 모든 문서들을 받아와서 복호화를 수행한 다음에 검색을 수행해야 한다. 이러한 비효율적인 것들을 막기 위해 암호화된 데이터베이스 검색 기술들이 연구되고 있다. 암호화된 데이터베이스 검색 기술은 크게 암호화 기법에 기반한 접근법[2-8]과 정보 은닉에 기반한 접근법[9-13]으로 나눌 수 있다.

암호화 기법에 기반한 접근법은 암호화 기법 자체에

검색 기능을 추가한 방법을 말한다. 이러한 방법을 사용하여 사용자들은 암호화된 데이터를 복호화하지 않고도 암호화된 상태로 다양한 검색 기능을 수행할 수 있다. 이러한 방법은 세부적으로 검색이 가능한 암호화 기법[2-6]과 순서 유지 암호화 기법[7-8]으로 나눌 수 있으며, 특히 검색이 가능한 암호화 기법은 대칭키 기반의 검색이 가능한 암호화 기법[2-4]과 공개키 기반의 검색이 가능한 암호화 기법[5-6]으로 나눌 수 있다. 암호화 기법에 기반한 접근법은 기존의 암호화 기법에 검색 기능을 추가한 형태이며, 증명을 통해 강한 안전성을 보장한다. 하지만 대용량 데이터베이스를 고려하여 설계되지 않고 각각의 암호문에 대한 검색을 위해 설계되었기 때문에 데이터베이스에서의 검색 효율성은 상대적으로 낮은 편이다.

위와 같은 문제점들을 해결하기 위해 대용량 데이터베이스를 위한 효율성 중심의 정보 은닉에 기반한 접근법이 연구되어 왔다. 이것은 원본 데이터는 안전성이 검증된 암호화 방식을 사용하여 암호화를 수행하고, 이와 별도로 추가적인 저장 공간을 사용하여 데이터에 대한 키워드들을 인덱스 형태로 저장하는 방법이다. 따라서 어떤 키워드를 사용하여 데이터를 검색하는 경우, 데이터베이스 전체를 검색하는 것이 아니라 인덱스 부분만을 검색하기 때문에 필요한 데이터를 효율적으로 검색할 수 있다는 장점을 가진다. 하지만 이러한 방법은 추가적인 저장 공간이 필요하다는 단점이 존재하며, 인덱스 생성 방법에 대한 안전성 증명이 이루어지지 않아 상대적으로 약한 안전성을 가진다. 이러한 방법에는 크게 버켓, 해시, 블룸 필터를 사용한 방법들이 있다. [표 1]은 암호화된 데이터베이스에 사용 가능한 검색 기법들을 분류한 것이다.

3장에서는 암호화 기법에 기반한 접근법에 대해서 검색 가능한 암호화 기법과 순서 유지 암호화 기법으로 나누어 살펴보고, 4장에서는 정보 은닉에 기반한 접근법에 대해서 버켓 기반 암호화 기법, 해시 기반 암호화 기법, 블룸 필터 기반의 암호화 기법으로 나누어 살펴본다.

〔표 1〕 암호화된 데이터베이스 검색 기법 분류

	암호화 기법에 기반	정보 은닉에 기반
기법	검색 가능 암호화, 순서 유지 암호화	버켓 기반 암호화, 해시 기반 암호화, 블룸 필터 기반 암호화
효율성	낮음	높음
안전성	높음	낮음

Ⅲ. 암호화 기법에 기반한 접근법

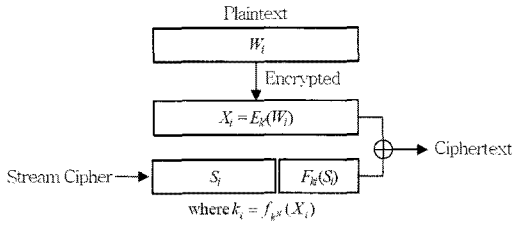
본 장에서는 암호화 기법에 기반한 접근법을 검색이 가능한 암호화 기법과 순서 유지 암호화 기법으로 나누어 살펴보며, 검색이 가능한 암호화 기법을 좀 더 세부적으로 대칭키 기반의 검색 가능 암호화 기법과 공개키 기반의 검색 가능 암호화 기법으로 나누어 살펴본다.

3.1 대칭키 기반의 검색 가능 암호화 기법[2,3]

대칭키 기반의 검색이 가능한 암호화 기법은 자신의 비밀키로 데이터를 암호화하여 서버에 저장하고, 이후에 이러한 데이터를 자신만이 검색 및 복호화할 수 있는 구조이며, 웹 스토리지 환경에서 주로 사용될 수 있는 기법이다.

2000년, D. Song 등은 검색이 가능한 암호화 기법에 대한 연구를 최초로 제안하였다[2]. 여기에서는 대칭키 기반의 의사난수 함수와 의사난수 생성기를 프리미티브(primitive)로 사용하여 검색 가능한 기법을 제안하였으며, 이것은 어떤 키워드에 대한 트랩도어(trapdoor)가 주어지는 경우, 신뢰하지 않는 서버가 암호문을 검색하여 그 키워드를 포함하고 있는지를 테스트할 수 있는 연구이다. [그림 1]은 이 논문에서 제안하는 기법을 간략하게 나타낸 그림이다.

평문인 W_i를 미리 정한 안전한 대칭키 암호화 기법 E_k를 사용하여 암호화하여 X_i를 생성한다. 의사난수 생성기를 사용하여 S_i를 생성하고, 일방향 함수 $f(x)$와 X_i를 사용하여 k_i를 생성하여 최종적으로 스트림 암호화 기법의 키로 사용될 $<S_i, F_{k_i}(S_i)>$를 생성한다. 이것을

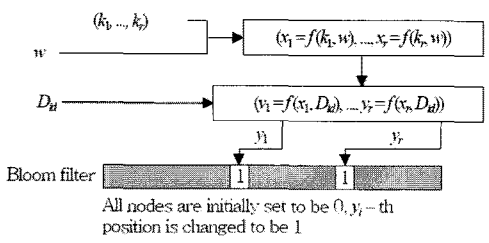


〔그림 1〕 Hidden Search 구조

대칭키 암호화 기법을 사용하여 생성된 X_i와의 XOR 연산을 통해 최종적인 암호문을 생성한다. 이렇게 생성된 암호문을 검색하기 위해서 사용자는 X_i와 k_i를 계산해야 한다. 이 기법은 대칭키 암호화 기법을 사용하여 평문의 노출을 완전히 차단하며, 스트림 암호화 기법의 구조를 통해 검색 기능을 추가한 구조를 가진다. 사용자는 사용된 대칭키 암호화 기법의 비밀키 및 의사난수 생성기의 시드(seed) 값을 알고 있기 때문에, 해당 키워드를 효율적으로 검색할 수 있다. 이 논문은 최초로 대칭키 기반의 검색이 가능한 암호화 기법이라는 점에서 의의가 있지만, 수용 가능한 안전성에 대한 증명이 부족한 단점이 있다.

2003년, E-J. Goh는 블룸 필터[14]를 사용하여 최초로 안전성이 증명된 검색 가능 암호화 기법을 제안하였다[3]. 블룸 필터는 1970년에 Burton H. Bloom에 의해서 고안된 확률적인 자료 구조이며, 주어진 집합에 특정 원소가 속해 있는지를 판단하는 데 사용된다. 블룸 필터에서는 해시 함수로 계산된 위치를 1로 치환하여 인덱스로 저장하는 방식을 취하고 있다.

사용자는 키워드 w를 암호화하기 위해 우선 상수 r을 정하고, $x_1 = f(k_1, w), \ldots, x_r = f(k_r, w)$를 계산한다. 여기서 $k_1, \ldots, k_r$은 각 사용자들의 비밀키이다. 다시 문서에 대한 식별자 D_{ID}를 사용하여 다음과 같이 $y_1 = f(x_1, D_{ID}), \ldots, y_r = f(x_r, D_{ID})$를 계산한다. 이렇게 생성된 r개의 값에 대응되는 블룸 필터의 위치를 1로 치환하는 방식을 통해 w에 대한 인덱스를 생성한다. 사용자가 특정한 키워드 w를 검색하고자 하는 경우, 위와 동일한 방식으로 함수 값을 생성하여 블룸 필터를 통해 특정 문서가 해당 인덱스를 포함하고 있는지를 식별하는 방법을 통해 검색할 수 있다. 하지만, 블룸 필터의 특성상 확률적인 오류가 발생할 수 있다. 즉, 포함된 문서는 반드시 검색이 되지만, 포함하지 않는 문서 역시 어느 정도의 확률로 검색이 되는 오류가 발생한다. 따라서 본 기법에서는 필터링 과정이 필요하기 때문에, 추가적인 연산이 발생하는 단점이 존재한다. 블룸 필터의 길이는 안전성과 효율성의 반비례 관계를 이룬다. 즉, 블룸필터의 길이를 늘리게 되면 중복되는 키워드의 개수가 감소하여 보다 정확한 검색이 가능하다. 즉, 이러한 경우에 효율성은 높아지지만 공격자에게 키워드에 대한 인덱스 정보를 많이 노출하게 되어 안전성이 감소하게 된다. 반대로 블룸 필터의 길이를 줄이면 보다 안전하지만, 중복 검색이 많이 발생하게 되어 필터링의 수가 증가하게 된다.



〔그림 2〕 Secure Index 구조

3.2 공개키 기반의 검색 가능 암호화 기법[5,6]

공개키 기반의 검색이 가능한 암호화 기법은 검색을 할 수 있는 사람의 공개키로 암호화하여 서버에 저장하며, 이것에 대응되는 비밀키를 가진 사람만이 데이터를 검색하고 복호화할 수 있는 기법이다. 이메일 서버가 가장 대표적인 환경이며, 메일을 전송할 때 상대방의 공개키를 사용하여 암호화를 하고, 수신자는 자신의 비밀키를 사용하여 서버에 저장된 메일을 검색 및 복호화할 수 있다.

공개키 기반의 검색 가능한 암호화 기법에 대한 연구는 2001년에 D. Boneh 등에 의해 제안된 ID-기반 암호화 기법[15]의 연구를 바탕으로 시작되었다. 이 연구는 겹선형 사상을 사용하여 설계되었으며, 이것의 정의는 다음과 같다.

겹선형 사상 $e: G_1 \times G_1 \rightarrow G_2$에 대해

1. *computable*: 임의의 두 원소 $g, h \in G_1$에 대해 $e(g,h) \in G_2$를 쉽게 계산할 수 있어야 한다.
2. *bilinear*: 임의의 두 정수 x, y와 임의의 두 원소 $g, h \in G_1$에 대해 $e(g^x, h^y) = e(g,h)^{xy}$을 만족한다.
3. *non-degenerate*: g가 G_1의 생성자이면, $e(g,g)$는 G_2의 생성자이다.

위의 세 가지 성질로 인해 겹선형 사상은 암호화 기법 설계에 매우 유용하게 사용된다. 특히, *bilinear* 성질은 서로 다른 두 개의 비밀값을 하나로 묶어주는 역할을 수행한다. 즉, 일반적인 그룹에서 매우 어려운 문제에 속하는 CDH(Computational Diffie- Hellman) 문제가 겹선형 사상에서는 쉽게 계산될 수 있다.

2004년, D. Boneh 등은 이러한 겹선형 사상을 사용하여 최초로 공개키 기반의 검색이 가능한 암호화 기법을 제안하였다[5]. 이것의 구조는 다음과 같다. 먼저, 위의 겹선형 사상과 동일하게 $e: G_1 \times G_1 \rightarrow G_2$를 정의하고, 비밀값 a를 선택하여 $h = g^a$로 설정한다. 데이터 D를 자신의 대칭키로 암호화하고, $peks(w) = [g^r, h_2(e(h_1(w), h^r))]$를 구성하여 인덱스처럼 함께 저장한다. w는 D에 해당하는 키워드 값이고, h_1, h_2는 각각 랜덤 오라클이다. 위와 같이 서버에 저장된 값을 사용자가 특정한 키워드 w를 포함하는 문서를 검색하고자 할 경우, 사용자는 트랩도어 $T_w = h_1(w)^s$를 생성하여 서버에게 전송한다. 서버는 $peks(w) = [A, B]$라 할 때, 모든 자료에 대해 $h_2(e(T_w, A)) = B$를 만족하는 데이터를 사용자에게 돌려준다. 이러한 공개키 기반의 키워드 검색은 pairing 연산을 데이터의 개수만큼 해야 한다는 비효율성이 존재하지만, 공개키 환경에서 최초로 제안된 기법이라는 점과, 공개키 기반의 검색이 가능한 암호화 기법에 대한 안전성 모델을 정립하고, 그것에 따른 안전성 증명을 증명하였다는 장점이 있다.

2004년 이후 공개키 기반의 검색 가능한 암호화 기법은 다양한 형태로 발전하였으며, 2008년 J. Katz 등에 의해 검색하고자 하는 키워드를 다항식 형태로 구성하여 좀 더 다양한 기능을 포함하여 검색할 수 있는 기법[6]이 제안되었다.

3.3 순서 유지 암호화 기법[7,8]

순서 유지 암호화 기법은 원본 데이터의 크기 순서를 암호화된 데이터에 그대로 반영하여 암호화하는 기법을 말한다. 순서 유지를 통한 암호화 방법은 다음과 같은 다양한 장점을 가진다. 첫째, 서로 다른 평문이 같은 암호문으로 대응되는 일이 발생하지 않아 일치 검색에 있어 추가적인 필터링 과정 없이 정확한 검색이 가능하다. 둘째, 데이터베이스의 업데이트가 매우 편리하다. 데이터들은 서로 독립적으로 암호화, 복호화가 진행되기 때문에 삭제하거나 삽입하는 경우 추가적인 연산이 필요하지 않다. 셋째, 암호화된 데이터베이스에서 범위 검색을 매우 효율적으로 가능하게 한다. 즉, 특정한 값 이상의 데이터들만 검색하고자 하는 경우, 사용자는 단순히 기준이 되는 값을 암호화하여 서버에 보내기만 하면 된다. 이 경우, 서버는 사용자에게 받은 값보다 크기가 큰 암호문만을 사용자에게 되돌려 주기만 하면 범위 검색이 된다. 이렇게 추가적인 복호화 작업 없이도 암호화된 데이터간의 크기 순서가 유지되기 때문에 위와 같은 작업들을 효율적으로 수행할 수 있다.

2003년, G. Ozsoyoglu 등은 의사난수를 사용한 순서 유지 암호화 기법을 제안하였다[7]. 사용자는 비밀 시드 값을 가지고 의사난수를 생성하여 평문의 크기만큼 난수 값을 더해주게 된다. 예를 들어, 10을 암호화하고자 하는 경우, 의사난수 생성기에서 생성된 난수 10개를 더한 값이 10에 대한 암호문이 되는 것이다. 이러한 방식은 매우 간단하지만, 암호화하고자 하는 숫자가 커질 경우 연산량이 매우 늘어나게 되는 단점이 있으며, 평문의 분포가 암호문에 그대로 드러나게 되는 치명적인 약점을 가지게 된다.

그리고 이 논문에서는 단조 증가하는 함수를 사용하여 순서 유지 암호화 기법 역시 제안하였다[7]. 단조 증가하는 함수의 결과값을 암호문으로 하는 단순한 구조를 갖기 때문에 이 역시 절차가 매우 간단하다. 하지만 함수의 특성상 n차 다항식의 경우, $n+1$개의 알려진 평문, 암호문 쌍으로 암호화 함수의 복원이 완전히 가능하다. 또한, 의사난수 생성기를 사용한 기법과 마찬가지로 평문의 분포를 숨기지 못하는 동일한 단점을 내포하게 된다.

2004년, R. Agrawal 등은 기존의 순서 유지 암호화 기법의 가장 큰 단점인 분포가 드러나는 문제를 해결하기 위해 사용자가 정한 분포에 맞게 암호문을 생성하는 방법을 제안하였다[8]. 모델링(modeling) 단계를 통해 타겟 암호문의 분포를 설정하고, 평탄화(flattening) 단계를 통해 평문의 분포를 동일하게 만든다. 마지막으로, 변환(Transformation) 단계에서 미리 설정한 타겟 분포에 맞게 암호문을 변환하게 된다. 이러한 작업을 통해 평문의 분포를 암호문에 전혀 드러나지 않게 수행할 수 있다.

순서 유지 암호화 기법은 암호문에서 평문의 크기 순서가 드러나는 설계 구조의 특성 때문에 CPA (chosen message attack) 이상의 안전성을 만족할 수 없다. 하지만 안전성을 적정수준으로 낮추는 대신 암호화된 데이터베이스에서 효율적인 범위 검색을 가능하게 하였다. 이러한 특징은 암호화 시스템의 안전성과 효율성의 반비례 관계를 표현하는 대표적인 예이다.

3.4 분석

대칭키, 공개키 기반의 검색 가능한 암호화 기법은 기존의 암호화 기법에 검색 기능을 추가한 형태를 가진다. 따라서 기존의 암호화 기법보다 더 많은 연산량이 요구된다. 논문 [2]는 대칭키 암호화 뿐만 아니라 스트림 암호화도 필요한 기법이고, 논문 [5]의 경우에는 검색하는 경우마다 paring 연산이 요구된다. 이러한 문제점을 해결하기 위해 순서 유지 암호화 기법이 제안되었다. 순서 유지 암호화 기법은 암호문에서 바로 범위 검색을 가능하도록 구성하였으나, 설계 구조 특성상 일정수준 이상의 안전성을 가지는 것은 불가능하다. 따라서 사용하고자 하는 환경에 적합한 기법의 선택이 요구된다.

Ⅳ. 정보 은닉에 기반한 접근법

본 장에서는 정보 은닉에 기반한 접근법을 버켓 기반의 암호화 기법, 해시 기반의 암호화 기법, 블룸 필터 기반의 암호화 기법으로 나누어 살펴본다.

4.1 버켓 기반의 암호화 기법[9]

버켓 기반의 암호화 기법은 버켓을 사용하여 인덱스를 생성하는 기법을 말한다. 이것은 인덱스를 사용한 검색 알고리즘에 대한 최초의 시도이며, 2002년에 H.

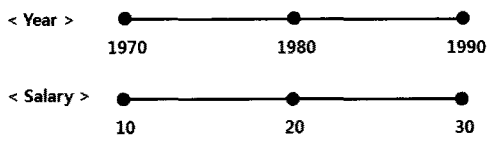
ID	Name	Year	Department	Salary
SA001	Alice	1981	Sales	20
MA001	Bob	1974	Marketing	30
RD001	Oscar	1983	R & D	25
MA002	Alice	1986	Marketing	20
RD002	Bob	1980	R & D	30
SA002	Mallory	1977	Sales	25

< Database >

Counter	Encrypted Tuple	I_1	I_2	I_3	I_4	I_5
1	S42te#^asse5	ε	θ	α	ξ	λ
2	Hhh346d#&d5	φ	μ	β	κ	δ
3	Kty%664&hyt	ε	μ	α	τ	λ
4	5GsdBNfg([2	φ	θ	α	κ	λ
5	[J786#@hrfmz	ψ	μ	α	τ	δ
6	90!@4her5(zx	ε	θ	β	ξ	λ

< Encrypted Database >

〔그림 3〕 버켓 기반 인덱스 생성의 예



〔그림 4〕 버켓팅의 예

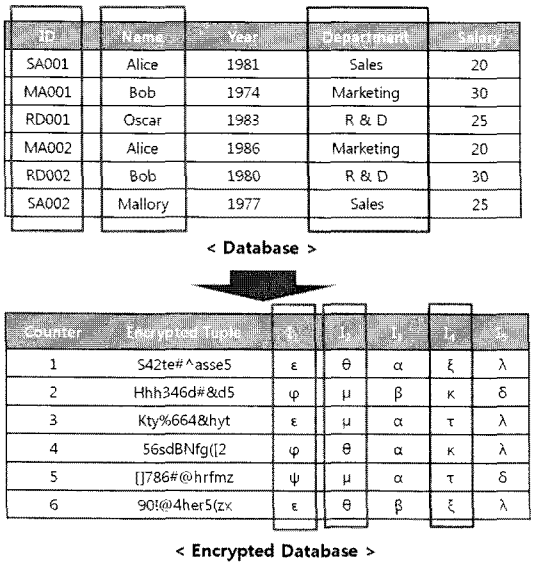
Hacigumus 등이 제안하였다. 버켓 기반의 암호화 기법은 평문의 어떤 속성에 대해 정의역의 범위를 겹치지 않게 구분하는 과정을 수행하여 대략적인 범위 검색을 가능하게 하는 알고리즘이다. 여기서 버켓은 속성의 정의역에 대해 균일하면서도 서로 겹치지 않게 나눈 것을 의미하며, 이렇게 나누는 과정을 버켓팅(bucketing)이라고 한다. [그림 3]은 버켓팅의 한 예를 나타낸다.

기존의 데이터베이스를 암호화하고 버켓팅을 하고자 하는 속성에 대해 추가적으로 버켓 기반 인덱스 생성 기법을 사용하여 데이터베이스에 인덱스 컬럼을 추가한다. 이러한 기법은 범위 질의가 부분적으로 가능하다는 특징도 가진다.

[그림 4]는 [그림 3]에서 버켓팅을 한 후에 데이터의 값이 동일한 버켓 내에 존재한다면, 동일한 버켓 ID를 할당하는 것을 나타낸 그림이다. 이러한 버켓 기반의 암호화 기법은 버켓팅을 어떻게 수행하는가에 따라서 검색의 정확성이 결정된다. [그림 3]에서와 같이 Year의 버켓 구간을 10으로 설정하면, 1980년에서 1990년에 속하는 모든 데이터는 동일한 문자로 인덱스가 생성되게 된다. 만약에 사용자가 1983년을 질의한다면, 테이블 내에서 1983년에 해당하는 값은 비록 하나뿐이지만, 사용자는 질의의 결과로 1,3,4,5번째 튜플(tuple)을 돌려받게 된다. 원하는 결과값만을 얻기 위해 버켓 기반의 암호화 기법은 서버로부터 결과값을 받은 이후에 추가적인 필터링 작업이 필요하다는 단점이 있다.

4.2 해시 기반의 암호화 기법[10,11]

해시 기반의 암호화 기법은 인덱스를 생성하는 경우에 해시 함수를 사용하여 암호화된 데이터베이스에서의 검색을 가능하게 하는 기법을 말한다. 인덱스를 생성하는 해시 함수를 선택할 때, 충돌성(collision)이 없는 해시 함수를 선택하게 되면, 정확한 일치 검색이 가능하게 되기 때문에 효율성 측면에서는 매우 뛰어나지만, 안전성 측면에서는 문제점을 가지게 된다. 그렇기 때문에 효율성을 어느 정도 낮추더라도 안전성을 높이기 위해 충



ID	Name	Year	Department	Salary
SA001	Alice	1981	Sales	20
MA001	Bob	1974	Marketing	30
RD001	Oscar	1983	R & D	25
MA002	Alice	1986	Marketing	20
RD002	Bob	1980	R & D	30
SA002	Mallory	1977	Sales	25

< Database >

Counter	Encrypted Tuple	A_0	I_0	I_1	I_2	I_3
1	S42te#^asse5	ε	θ	α	ξ	λ
2	Hhh346d#&d5	φ	μ	β	κ	δ
3	Kty%664&hyt	ε	μ	α	τ	λ
4	56sdBNfg([2	φ	θ	α	κ	λ
5	[]786#@hrfmz	ψ	μ	α	τ	δ
6	90!@4her5(zx	ε	θ	β	ξ	λ

< Encrypted Database >

〔그림 5〕 해시 기반 인덱스 생성의 예

돌성이 있는 해시 함수를 선택한다.

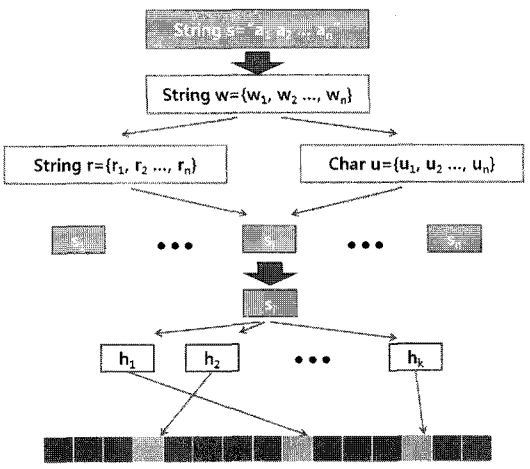
위의 [그림 5]는 해시 기반의 암호화 기법을 사용하여 인덱스를 생성한 테이블의 예시이다. 해시 기반 암호화 기법에서 해시 함수를 선택하는 경우, 충돌성을 가지는 해시 함수를 선택하기 때문에 질의하는 사용자가 원하지 않은 결과를 얻을 수도 있다. 예를 들어, 사용자가 ID 항목에서 MA001의 값을 가지는 사용자에 대한 정보를 요청했다고 하자. 그러면 그 질의에 대한 인덱스는 φ가 되며, 서버는 2,3,5번째 튜플을 사용자에게 검색의 결과로 되돌려주게 된다. 하지만 이러한 값들 중에서 실제로 ID 항목 중에서 MA001인 것은 2번째 튜플밖에 없기 때문에, 질의한 사용자는 자신이 검색한 값 이외의 값들도 받게 된다. 이러한 값들을 제거하기 위해 사용자는 추가적인 필터링 작업을 수행해야 한다.

이러한 해시 기반의 암호화 기법은 버켓 기반의 암호화 기법과 마찬가지로 일치 검색과 범위 검색을 수행하는 경우, 추가적인 필터링 작업이 필요하다는 단점이 있다. 하지만 해시 기반의 암호화 기법은 버켓 기반의 암호화 기법과 마찬가지로 숫자 데이터나 문자 데이터에 관계없이 사용이 가능하다는 장점도 존재한다. 또한, 버켓 기반의 암호화 기법과는 달리 해시 기반의 암호화 기법은 일방향 해시함수를 사용한다는 특징이 있다. 이것은 해시 함수를 사용할 때 충돌성이 없는 해시 함수를 사용하게 되면 정확한 일치검색이 가능하게 되지만 유추 공격에 취약하게 되며, 충돌성이 있는 해시 함수를 사용하게 되면 검색 결과에 추가 필터링 작업이 필요하게 되지만 랜덤하게 동일한 값을 부여하므로 유추 공격에 대한 안전성을 높일 수 있다는 특징이 있다.

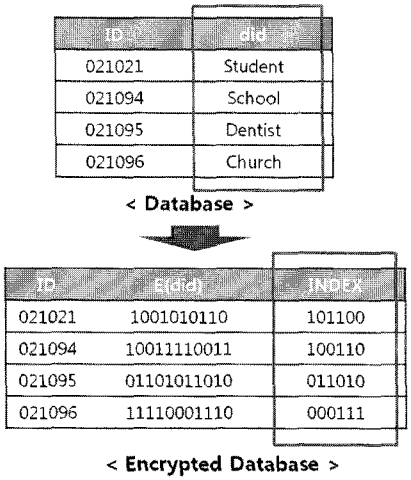
4.3 블룸 필터 기반의 암호화 기법[12,13]

블룸 필터 기반의 암호화 기법은 앞의 3.1절에서 살펴본 블룸 필터에 기반하여 설계된 암호화 기법을 말한다. 아래의 [그림 6]은 문자열 s를 k개의 해시함수를 통해서 블룸 필터로 구성된 하나의 예시이다.

위의 [그림 6]과 같이 문자열 s에 대해서 triple (w,r,u)을 구성하고, 이것을 각각의 비트열로 표현하여 s_1에서 s_n까지의 입력값을 만든다. 이것을 k개의 해시 함수에 통과시켜서 출력되는 값에 해당하는 비트를 1로 바꿔준다. 모든 s_i에 대해서 이러한 과정을 수행하고 나면, 결과적으로 길이가 m인 블룸 필터를 구성할 수 있다. 물론 블룸 필터의 길이 m을 설정할 때 작은 값으로 선택하게 되면 보다 많은 충돌이 일어나게 되어 원하는 값이 저장되어있지 않는 경우에도 결과값이 생기는 경우가 발생하는 단점이 존재한다. 따라서 블룸 필터를 사용하고자 하는 경우에는 안전성과 효율성의 정도에 맞는 적절한 블룸 필터의 길이를 선택하는 것이 중요한 문제이다. 아래의 [그림 7]은 블룸 필터 기반의 암호화 기법을 사용하여 인덱스를 생성하는 예시이다.



〔그림 6〕 블룸 필터 생성의 예



〔그림 7〕 블룸 필터 기반 인덱스 생성의 예

4.4 분석

4.4.1 버켓 기반의 암호화 기법

버켓 기반의 암호화 기법은 버켓팅을 통해서 형성된 구간과 버켓 ID와의 대응 함수를 잘 정의하면 숫자 데이터와 문자 데이터에 대해 모두 적용이 가능하다는 장점이 있고, 복잡한 데이터도 버켓팅을 통해서 비교적 간단하게 표현이 가능하다는 장점도 있다. 하지만 일치 검색의 경우에 정확하게 일치하는 평문을 알아내기 위해서는 추가적인 필터링 작업이 필수적이다. 범위 검색의 경우에는 범위 내의 값이 포함된 버켓을 검색하고, 이어서 버켓 내의 값을 모두 복호화해야 하는 과정이 필요하므로 실질적으로는 범위 검색이 이루어진다고 볼 수 없다. 그리고 버켓의 정보는 알려진 평문 공격(known plaintext attack)에 취약하므로 버켓 기법을 통해서 생성한 인덱스는 안전성을 보장받을 수 없다.

4.4.2 해시 기반의 암호화 기법

해시 기반의 암호화 기법은 버켓 기반의 암호화 기법에 비하여 조금 더 높은 안전성을 지닌다고 할 수 있다. 해시 기반 암호화 기법에서는 충돌성이 있는 해시 함수를 사용하기 때문에 서로 다른 값에 대해서도 동일한 출력값을 가질 수 있다. 해시 기반의 암호화 기법은 높은 안전성을 제공하기는 하지만 검색의 효율성 측면에서는 상대적으로 낮다고 할 수 있다. 이러한 안전성과 효율성을 적절한 수준에서 만족시키기 위해서는 해시 함수의 출력 범위를 사용 환경에 맞게 적절하게 선택해야 한다. 반면, 해시 함수는 데이터를 랜덤한 값으로 생성해주기 때문에 범위 검색은 지원하지 않는다.

4.4.3 블룸 필터 기반의 암호화 기법

블룸 필터 기반의 암호화 기법은 해시 함수를 이용한다는 측면에서 해시 기반의 암호화 기법과 어느 정도 유사하다고 볼 수 있다. 블룸 필터 기반의 암호화 기법에서 사용자가 데이터를 검색하는 경우, 어느 정도의 실패할 가능성이 존재한다는 점과 안전성과 효율성을 적정 수준에 맞게 선택하기 위해서는 해시 함수의 출력 범위를 사용 환경에 맞게 적절하게 선택해야 한다는 공통점도 존재한다. 하지만 해시 기반의 암호화 기법처럼 해시 함수의 출력값을 직접적이 아닌 간접적으로 활용한다는 측면에서 차이점이 있다고 할 수 있다.

V. 결 론

본 고에서는 스마트그리드 환경에서 발생 가능한 보안 위협에 대해 살펴보고, 스마트그리드 환경에 적용이 가능한 암호화된 데이터베이스 검색 기술에 대해 비교 및 분석하였다. 암호화 기법에 기반한 접근법은 높은 안전성을 제공하지만, 검색하는 과정이 상대적으로 비효율적이다. 반면, 정보 은닉에 기반한 접근법의 경우 데이터를 검색하는 과정은 효율적이지만, 상대적으로 높은 안전성을 보장하지는 못한다. 이러한 기존의 연구들은 데이터베이스를 사용하는 환경에 일반적으로 적용이 가능한 검색 기술들이며, 스마트그리드 환경에서 역시 적용이 가능하다. 하지만 이러한 기술들은 모두 안전성과 효율성의 상충관계를 가지며, 따라서 스마트그리드 환경에 특화시켜 적용되는 환경에서 안전성 및 효율성을 극대화시키는 연구 또한 필요하다.

참 고 문 헌

[1] 윤인하, "최근 미국 동부지역의 정전사태와 미국 전력 사업의 문제점", Aisa-Pacific Review, 2003.

[2] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searching on Encrypted Data,"

IEEE Symp. on Security and Privacy, May 2000.

[3] E.J. Goh, "Secure Indexes," Technical Report 2003/216, IACR ePrint Crpytography Archive, June 2003.

[4] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data," In Applied Cryptography and Network Security Conference, June 2004.

[5] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Eurocrypt 2004, May 2004.

[6] J. Katz, A. Sahai, and B. Waters, "Predicate Encryption Supporting Disjunctions," EUROCRYPT 2008, LNCS 4965, pp. 146-162, 2008.

[7] Gulteckin Ozsoyoglu, David Singer, "Anti-tamper Database: Querying Encrypted Databases," In Proc. of the 17th Annual IFIP WG 11.3 Working Conference on Database and Applications Security, Estes Park, Colorado, August 2003.

[8] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu, "Order Preserving Encryption for Numeric Data," In Proc. of the ACM SIGMOD Conf. on Management of Data, Paris, France June 2004.

[9] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," ACM SIGMOD, 2002.

[10] E. Damiani, S.D.C.di Vimercati, S. Jajodia, S. Paraboshi and P. Samarati, "Balancing Confidentiality and Efficiency in Untrusted Relational DBMSS," In Proc. of the 10th ACM Conf. on Computer and Communications Security(CCS), October 2003.

[11] A. Ceselli, E. Damiani, S.D.C.D Vimercati, S. Jajoda, S. Paraboschi and P. Samarati, "Modeling and Assessing Inference Exposure in Encrypted Database," ACM Transactions on Information and System Security, Vol.8, No.1, Feb. 2005.

[12] L. Liu, J. Gai, "Bloom Filter Based Index for Query over Encrypted Character Strings in Database," csie, vol. 1, pp.303-307, 2009 WRI World Congress on Computer Science and Information Engineering, 2009.

[13] L. Liu and J. Gai, "A Method Query over Encrypted Data in Database," ICCET, Volume 01, pages 23-27, 2009.

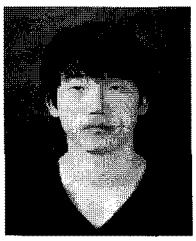
[14] M. Mitzenmacher. "Compressed Bloom Filter," IEEE/ACM Transactions on Networking, vol. 10, no. 5, pp. 604-612 December 2002.

[15] D. Boneh, M. Franklin, "Identity-Based Encryption from the Weil Pairing," CRYPTO 2001, August 2001.

〈著者紹介〉

김 기 성 (Gi Sung Kim)
학생회원

2009년 2월 : 서울시립대학교 수학과 졸업
2009년 3월~현재 : 고려대학교 정보경영공학전문대학원 석사과정
관심분야 : 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론

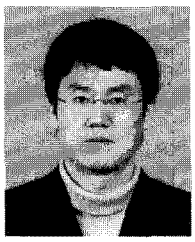


김 동 민 (Dong Min Kim)
학생회원

2009년 2월 : 서울시립대학교 수학과 졸업
2009년 3월~현재 : 고려대학교 정보경영공학전문대학원 석사과정
관심분야 : 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론

노 건 태 (Geon Tae Noh)
학생회원

2008년 2월 : 고려대학교 산업시스템정보공학과 졸업
2010년 2월 : 고려대학교 정보경영공학전문대학원 석사 졸업
2010년 3월~현재 : 고려대학교 정보경영공학전문대학원 박사과정
관심분야 : 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론



정 익 래 (Ik Rae Jeong)
정회원

1998년 2월: 고려대학교 전산학과 학사 졸업
2000년 2월: 고려대학교 전산학과 석사 졸업
2004년 8월: 고려대학교 정보보호대학원 박사 졸업
2006년 6월 ~ 2008년 2월: 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월 ~ 현재: 고려대학교 정보경영공학전문대학원 조교수
관심분야 : 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론