스마트 그리드에서 향상된 Homomorphism을 이용한 안전한 데이터 애그리게이션

박지혜, 최경, 도인실, 채기준
이화여자대학교 컴퓨터공학과
e-mail : elite_pjh@naver.com, cbk0907@ewhain.net, isdoh@ewhain.net, kjchae@ewha.ac.kr

Secure Data Aggregation Using Enhanced Homomorphism in Smart Grids

Jihae Park, Kyung Choi, Inshil Doh, Kijoon Chae
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

최근 그린 IT 에 대한 관심이 점차 고조되면서, 이 사업의 일환으로 저탄소, 녹색 성장을 위한 지능형 전력망인 스마트 그리드의 도입이 발 빠르게 진행되고 있다. 스마트 그리드를 통해 전력 공급자와 소비자의 양방향 통신으로 에너지 효율의 최적화가 가능하지만, 동시에 사이버 공격에 의한 개인정보의 노출위험에 대한 우려도 제기되고 있는 상황이다. 데이터의 안전한 전송을 위해 다양한 암호화 방식이 제안되고 있으며, 본 논문에서는 기밀성에만 초점을 맞춘 Homomorphic 암호화의 허점을 보완하기 위하여 additive Homomorphic 방식을 기반으로 하여 데이터 무결성을 보장할 수 있는 새로운 방식을 제안하였다. 이 메커니즘을 통해 데이터는 최종 목적지까지 안전하게 도달했으며, 위조 및 변조 되지 않았다는 것을 보장받을 수 있다.

1. 서론

스마트 그리드는 기존의 전력망에 정보통신 기술을 접목하여 전력 공급자와 소비자가 양방향으로 실시간 정보를 교환함으로써 에너지 효율을 최적화하는 차세대 지능형 전력망이다.

스마트 그리드 기술은 소비자의 에너지 사용을 실시간으로 감시하고, 가정용 기기와 통신함으로써 에너지 효율을 가정으로까지 확대해 준다. 그러나 스마트 그리드에서 수집된 정보는 소비자의 개인 정보를 크게 해칠 우려가 있기 때문에 각 가정의 스마트 미터기로부터 최종 게이트웨이까지의 안전한 전송이 보장되어야 한다.

본 논문에서는 Homomorphic 암호화를 통한 안전한 데이터 전송을 기반으로 하며, 단순한 modular 연산의 특징을 이용하여 데이터의 위조 및 변조 여부를 확인할 수 있는 방안을 제시하고자 한다.

2. 관련 연구

2.1 Homomorphic 암호화

Homomorphic 암호화는 평문에 대한 특정 연산이 암호문에 대한 연산에 의해 수행되는 특징을 가지고 있다. 이에 따라 중간 과정에서 평문이 직접 드러나지 않게 되므로 기밀성을 보장하기 위한 안전한 암호화 방식이라고 볼 수 있다.

즉 Homomorphic 암호화 함수를 E() 라 하고, 두 개의 메시지 $x, y \in Z_N$, 를 가정했을 때, 평문 x, y 나 비밀키(K_1, K_2)를 알지 못하고도 $E_K (x \bigstar y) = E_{K1} (x) \circ E_{K2} (y)$ 를 계산하는 것이 가능하다. 실제적으로 ★ 연산은 덧셈 또는 곱셈으로 이루어져 있다.

이러한 Homomorphic 속성은 개인 정보 검색 전략 및 충돌 방지 해시 함수, 안전한 투표시스템을 만드는 데 사용될 수 있다.

널리 알려진 Homomorphic 암호화로는 RSA[1], EL Gamal[2], Paillier[3], Naccache-Stern[4], Boneh-Goh-Nissim(BGN)[5] 등이 존재하고 데이터 애그리게이션을 위해서는 additive Homomorphic 원리가 요구되며, 일반적으로 Boneh-Goh-Nissim(BGN) 과 Paillier 방식이 주로 사용된다.

2.2 BFS-Spanning tree

BFS-Spanning tree 는 BFS(Breath First Search: 너비우선탐색) 알고리즘을 이용한 스패닝트리 구성방식이다. 이 방식은 먼저 루트노드로부터 주변의 가능한 모든

이 논문은 2009 년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(NRF-2009-0083985)

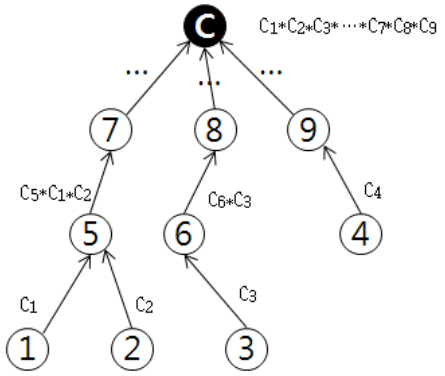
노드를 탐색하고, 그 다음 노드로 이동하여 사이클을 발생시키는 경로를 제외한, 그 이외의 모든 경로를 탐색하는 방식으로 점점 넓게 범위를 확장시키면서 모든 노드가 하나의 스패닝트리를 구성하게 된다.

다른 방식의 스패닝트리 구성에 비해 BFS 방식을 이용하면 트리의 구성이 상대적으로 낮고 넓게 형성되므로 애그리게이션과정의 시간을 단축시킬 수 있다.

2.3 Additive Homomorphic Aggregation

2.2 에서 언급했던 BFS 방식으로 트리의 구성이 완료되면, 각 노드에서 발생하는 측정값들에 대하여 애그리게이션 과정이 일어나게 된다.

Homomorphic 방식의 특성을 이용하게 되면, 각 노드에서는 자식노드로부터 전달받은 데이터들과 자신의 측정치를 암호화한 값을 단순하게 곱하여 부모노드로 전송하게 된다[3].



[그림 1] 기밀성을 위한 애그리게이션 과정

예를 들어, [그림 1]과 같이 9 개의 노드로 구성되어 있는 네트워크를 가정한다면 위에서 언급한 방식에 의해 collector 가 받은 최종적인 C_{col} 값은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} C_{col} &= C_{o7} * C_{o8} * C_{o9} \\ &= (C_{p7} * C_{o5})*(C_{p8} * C_{o6})*(C_{p9} * C_{p4}) \\ &= C_{p7} * (C_{p5} * C_{p1} * C_{p2}) * C_{p8} * (C_{p6} * C_{p3}) * \\ &\quad C_{p9} * C_{p4} \\ &= C_{p1} * C_{p2} * \ldots * C_{p8} * C_{p9} \end{aligned}$$

C_{oi} : i 번째 노드까지의 애그리게이션 결과,
C_{pi} : i 번째 노드에서의 측정치를 암호화한 값

결과적으로 collector 에게 모인 값 C_{col} 은, 모든 노드들로부터 발생한 측정치 값이 각각 암호화되어 곱해진 것과 동일하게 된다.

따라서 additive Homomorphic 원리에 의해 암호문들의 곱을 복호화 함으로써 평문들의 합을 얻어낼 수 있다.

$$D(C_{col}) = P_1 + P_2 + \ldots + P_8 + P_9$$

D(): 복호화 함수
P_i: 각 노드에서의 측정값

3. 제안 모델

앞서 설명한 Homomorphic 암호화의 원리에 의해, 중간 노드에서 평문의 노출이 발생하지 않으므로 기밀성이 보장될 수 있다.

하지만 악의적인 injection 공격을 통해 실제 값과 다른 평문으로 복호화되도록 변조된 암호문을 생성하는 것이 가능하다.

그 결과, 최종적으로 collector 에게 정확하지 않은 애그리게이션 결과가 전송되게 되므로 변조 여부를 확인하는 메커니즘이 추가 되어야 할 필요가 있다.

본 논문에서는 Homomorphic 암호화의 허점을 보완하기 위하여, Homomorphic 방식의 암호화 데이터 전송을 기반으로 한 추가적인 무결성 검증방식을 제안하였으며, 이는 다양한 additive Homomorphic 암호화 방식에 대하여 독립적으로 적용될 수 있다.

무결성 검증을 위해 각 노드에서 Z_i 값을 추가로 계산하여 전송함으로써 collector 에 최종적으로 전송된 $\sum Z_i$ 값을 통해 데이터의 변조여부를 확인할 수 있는 방안을 제시한다.

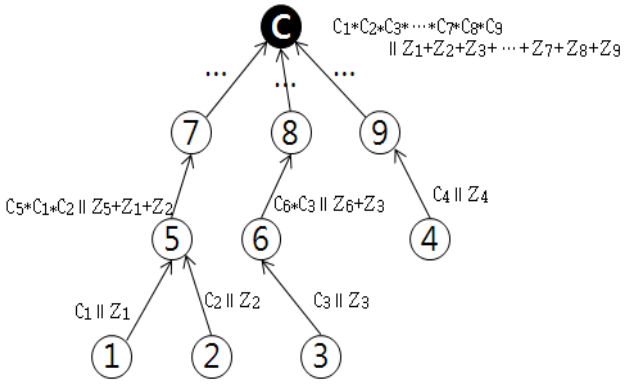
Z_i 는 다음과 같은 식으로부터 유도된다.

$$Z_i = P_i + X * count_i \pmod r$$

- P_i : 각 노드에서의 측정값
- $count_i$: 연결된 하위 레벨의 총 노드 수
- r : 트리 구성 시 마다 collector 가 생성하며, $r > M*N$ 을 만족하는 랜덤한 수 (M: 최대허용전력, N: 총 노드 수)
- X : r 보다 크며, r 의 배수가 아닌 임의 값

Count 값은 트리 구성 시 모두 0 으로 초기화 되어 있다. 부모 노드에게 1 증가 시킨 값을 전송하며, 부모노드는 자식노드들로부터 받은 count 값들을 합하여 자신의 count 값을 갱신한 후, 자신의 부모노드로 전송한다. 최종적으로 collector 에게 전송된 count 값은 총 노드수가 된다.

주기적으로 트리가 재구성 될 때마다, collector 는 r 값과, 조건에 맞는 X 를 생성한 후 안전하게 공유되는 링크 키를 통해 네트워크상의 모든 노드에게 전송한다.



[그림 2] 무결성을 추가한 애그리게이션 과정

[그림 2]는 [그림 1]과 달리 무결성을 위해 Z 라는 새로운 값이 추가 전송되는 것을 보여준다.

새로운 메커니즘 제안을 위해, 임의적으로 additive Homomorphic 방식 중 Paillier 방식으로 데이터 애그리게이션이 일어나며, 노드간의 네트워크 구성 방식은 기존에 제안되었던 BFS(Breath First Search) -Spanning tree 기반으로 주기적으로 일어난다고 가정하였다[6]. 트리 구성이 완료되면, 공유되는 링크 키를 이용하여 collector 와 모든 노드들은 r 값과 X 값을 공유하게 된다.

그 이후에 [그림 2]에서 볼 수 있듯이 각 노드에서는 두 가지 연산이 수행되어 결합된다. 첫 번째는 Homomorphic 암호화를 이용한 데이터 기밀성 보장을 위한 곱셈연산과, 두 번째는 무결성 보장을 위한 Z 값들의 덧셈연산이다. 이 두 연산은 상향식으로 최종적으로 collector 까지 전송되며, 각 노드는 자신의 암호화된 측정값과 자식 노드들로부터 전달받은 암호문들을 모두 곱하고, 자신의 Z 값과 자식 노드로부터 전달받은 Z 값을 모두 더한 후 두 값을 결합하여 자신의 부모 노드에게 전송한다.

따라서 최종적으로 collector 에게 전송된 값은 다음과 같다.

$$C_{col} \parallel Z_{col} =$$
$$C_1*C_2*C_3*\cdots*C_7*C_8*C_9 \parallel Z_1+Z_2+Z_3+\cdots+Z_7+Z_8+Z_9$$

또한, Z_{col} 값은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} Z_{col} = {} & P_1 + X * count_1 \pmod r + \\ & P_2 + X * count_2 \pmod r + \\ & \cdots \\ & P_8 + X * count_8 \pmod r + \\ & P_9 + X * count_9 \pmod r \\ = {} & (P_1+P_2+ \ldots +P_8+P_9) + (X * \sum count_i) \pmod r \end{aligned}$$

Collector 는 Homomorphic 암호화 원리에 의해 C_{col} 을 복호화함으로써 $P_1+P_2+ \ldots +P_8+P_9$ 를 얻을 수 있으며, r 과 X, 전체 노드 수에 대해 알고 있으므로 $(D(C_{col}) + X * \sum count_i) \pmod r$ 를 계산하여 Z_{col} 과 같은지 확인한다.

중간 노드들을 거치는 동안 악의적인 공격자에 의해 원본과 다르게 복호화 되도록 암호문이 변조 되었다면 $D(C_{col})$는 $\sum P_i$ 와 서로 다른 값을 갖게 되고, 결국 Z_{col} 값과 다른 값을 생성하게 된다.

이를 통해 최종 데이터의 변조 여부를 판단할 수 있으며, 데이터 변조가 일어났다면 collector 는 무결성이 보장되지 않은 데이터들을 즉시 폐기하고 다시 스패닝트리를 재구성 하여 동일한 애그리게이션 과정을 반복하게 된다.

본 논문에서 제안한 방법은 modular 연산의 특성상, 변조된 이후에도 동일한 modular 값을 갖게 되는 경우가 발생할 수도 있다. 따라서 이를 방지하기 위해 r 값을 $r > M*N$ 을 만족하는 랜덤 한 수(M: 최대허용전력, N: 총 노드 수)로 설정하였다.

$D(C_{col})$ 과 $P_1+P_2+P_3+\ldots+P_9$ 가 동일한 modular 값을 갖기 위해서는 두 값이 동일하거나, r 의 배수만큼 차이가 나야 한다.

따라서, 두 값의 차이가 r 의 배수가 될 수 없도록 설정하게 되면 서로 다른 두 값에 대하여 동일한 modular 값이 발생하는 것을 막을 수 있다.

각 가정에서 전력 사용량이 최대 허용 전력을 넘어가게 되면 전력 과부하로 인해 차단기가 내려가게 되므로, $\sum P_i$ 는 모든 노드가 최대 허용 전력값을 갖게 되는 M*N 이상의 값을 가질 수 없다.

따라서, r 값을 현실적으로 측정될 수 없는 값인 M*N 이상의 값으로 설정하게 되면, 변조된 전력량의 합은 항상 r 보다 작고, r 의 양의 배수가 될 수 없으므로 결론적으로 서로 다른 두 값에 대해 동일한 modular 값이 나오게 되는 것은 불가능하게 된다.

본 논문에서 제시한 무결성 검증방법은 간단한 덧셈, 곱셈, modular 연산만으로 데이터의 변조여부를 확인할 수 있다. 또한 r, X, $count_i$ 값은 스패닝트리가 생성될 때마다 매번 다른 값을 생성하게 된다. 따라서 각 노드는 같은 측정값에 대하여 서로 다른 Z_i 값을 생성하게 됨으로써 Z_i 값을 통해 평문을 유추해내는 것을 방지할 수 있다.

4. 결론 및 향후 연구

최근 그린 IT 에 대한 관심이 높아지면서, 저탄소 녹색성장의 완성을 위해 지능형 전력망인 스마트 그리드 기술이 한발 앞서서 빠르게 발전하고 있다. 스마트 그리드를 통해 전력망이 진화하고 있지만 사이버 공격에 의한 개인정보의 노출위험에 대한 우려도 제기되고 있는 상황이다. 이에 따라 다양한 방법의 암호화 모델이 제시되고 있으며, 그 중 Homomorphic 암호화 방식은 최종 목적지까지 가는 도중에 암·복호화 과정을 거치지 않기 때문에 계산에 드는 비용 면에서도 효율적일 뿐 아니라 평문이 그대로 드러나는 것을 막음으로써 기밀성을 보장할 수 있다.

하지만 오로지 기밀성에만 강점을 보이며, 전송과정에서 악의 있는 공격자가 원본 데이터와 다르게 복호화 되도록 암호문 자체를 변경시킬 수 있는 가능성에 대해서는 고려하지 않았다.

따라서 본 논문에서는 암호문에 Z_i 라는 새로운 계산 값을 추가로 붙여서 전송함으로써 최종 데이터의 무결성을 보장할 수 있는 방안을 제시하였다. 매우 간단한 덧셈, 곱셈, modular 연산만을 이용하여 데이터의 위조 및 변조 여부를 판단하는 것이 가능하다.

향후 연구로는 제안 메커니즘의 시뮬레이션을 수행하여 가장 일반적인 무결성 인증 방식인 MAC 과 비교함으로써 보안성 및 효율성을 분석하고 이를 통해 본 메커니즘의 효율성을 입증하고자 한다.

5. 참고문헌

[1] R.L. Rivest, A. Shamir, and L. Adleman," A Method for Obtaining DigitalSignatures and Public-Key Cryptosystems", in Communications of the ACM, 1978.

[2] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in Proceedings of CRYPTO 84 on Advances in cryptology.

Springer-Verlag New York, Inc., pp. 10–18, 1985.
[3] P. Paillier, “Public-key cryptosystem based on composite degree residuosity classes,” in Proceedings of Eurocrypt ’99, 1999.
[4] D. Naccache and J. Stern, “A new public key cryptosystem based on higher residues,” in CCS ’98: Proceedings of the 5th ACM conference on Computer and communications security, pp. 59–66, 1998.
[5] D. Boneh, E. Goh, and K. Nissim, “Evaluating 2-dnf formulas on ciphertexts,” in Proceedings of Theory of Cryptography (TCC), pp. 325–341, 2005.
[6] Fengjun Li, Bo Luo, Peng Liu, “Secure Information Aggregation for Smart Grids Using Homomorphic Encryption” Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, 2010.