데이터 중복 제거 기술 분석

조민정*, 이창훈*
*서울과학기술대학교 컴퓨터 공학과
e-mail:whalswjd2518@gmail.com

An analysis of Data Deduplication techniques

Min-Jeong Jho*, Chang-hoon Lee*
*Dept of Computer Engineering, Seoul National Science and Technology

요 약

저장하는 데이터의 용량이 증가함에 따라 데이터들은 효율적으로 보관될 필요성이 증가하였다. 이에 따라, 데이터 용량을 줄이는 기술로 많은 서비스들이 데이터 중복 제거 기술을 사용한다. 본 연구에서는 일부 서비스의 데이터 중복 제거 기술을 분석하고 , 데이터 중복 제거 기술의 발전 동향을 예측하고자한다.

1. 서론

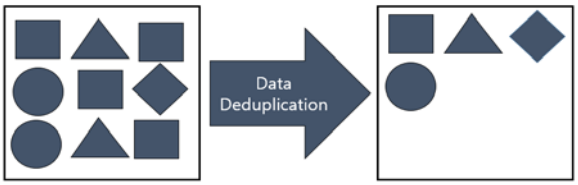
최근 사진 및 문서 등 저장이 필요한 데이터의 용량이 증가하고 있으며 이에 따라 대용량 저장 공간과 클라우드 서비스를 찾는 사람이 늘고 있다. 많은 데이터를 저장하기 위해 데이터의 용량을 줄이는 기술이 필요하게 되었다. 중복 제거 또한 그러한 기술들 중 하나이다. 중복 제거는 백업되는 데이터를 크게 줄일 수 있고, 스토리지 용량, 공간 그리고 에너지 소비를 줄일 수 있다[1].

본 논문에서는 평문 데이터를 중복 처리하는 서비스들을 대상으로 하여 중복 처리 기술을 살펴본다. 이 서비스들을 중복제거를 암호화 및 키 관리 관점에서 본다. 중복 제거에 대한 적용 기술에 대해 분석하고, 이에 관련된 향후 전망을 제시한다.

2. 중복 제거의 개념 및 관련 연구

1) 중복 제거의 개념

중복 제거는 같은 데이터를 업로드한 모든 사람을 데이터 소유자를 만들고 중복 데이터의 단일 데이터만을 공유한다. 중복제거는 스토리지 내의 중복되는 복사본들을 제거하는데 특화되어 있다[2].



(그림 1) 중복 제거

2) 관련 연구

1989년 R.C. Merkle등은 Merkle-Tree를 제안하였다. Merkle-Tree는 먼저 전체 데이터를 고정길이를 갖는 블록 단위로 분할한다. 그리고 인접한 두 블록의 해쉬 값을 하나의 해시 값이 출력할 때까지 계산한다[20].

2002년 Douceur 등은 평문을 파일을 해쉬해서 얻은 값을 키로 이용하여 데이터를 암호화하는 Convergent Encryption기법을 제안하였다[16].

2010년 Harnik등은 Convergent Enryption를 이용하여 저장할 데이터를 암호화하고, 시스템에 설정된 임계치만큼 반복하여 데이터를 저장하는 기법을 제안하였다[3].

2011년 Halevi등은 Merkle-Tree를 기반으로한 소유권 증명 프로토콜로 proofs of onwership(PoW)를 제안하였다. 이때 이 증명 프로토콜은 평문 데이터에 관한 것이다[20].

2013년 Mihir Bellare등은 먼저 Message-Locked Encryption(MLE)을 제안하였다. MLE는 Convergent Encryption의 악의적인 사용자에 의해 우연히 다른 메시지로 암호화 된 암호문과 같은 암호문이 존재할 경우, 먼

저 저장된 암호문에 대한 정보만 저장되어 이후 사용자의 접근이 차단되는 문제를 막기 위해서 Public 파라미터로 키를 생성하는 방법이다[21]. 이후 DupLESS를 제안하였고 이는 client에 연관된 그룹이 데이터를 Storage Service(SS)로부터 분리된 key server(KS)의 도움을 받아 데이터를 암호화하는 방식이다[17]. 이때 KS는 RSA-OPRF 프로토콜을 이용해 키를 분배한다[18].

2016년 김원빈등은 블룸필터를 이용하여 소유권 증명을 하였다. 이는 Merkle Tree 방식의 소유권 증명보다 빠르고 낮은 연산량을 갖는다[22].

3. 중복 제거 서비스 기술

1) Tivoli Storage Manager

IBM사의 Tivoli Storage Manager에서는 client-side data deduplicaton과 server-side data deduplication 2가지가 있다. 이때 중복제거는 블록 단위로 하며 평균 100KB이다.

clinet-side data deduplication은 클라이언트가 다른 파일과 비교하여 중복 여부를 확인할 수 있는 extent를 생성한다. 클라이언트와 서버는 중복 확인 가능한 extent와 같이 작동한다. 클라이언트는 서버에 중복되지 않은 extents를 보낸다. 이 후 전체 데이터의 블록마다 새로운 extents를 만들어 낸다. 이 extents의 일부 혹은 전부는 이전에 데이터 중복 제거 작동에서 생성된 extents와 일치하거나 server로 보내지게 된다. 일치하는 것은 서버로 다시 보내지 않는다[4].

server-side data deduplication은 2가지 단계가 있다. 첫 번째 단계는 서버에서 중복 데이터를 확인하고 두 번째 단계에서는 중복된 데이터가 서버의 처리 과정에 의해서 제거가 된다. 이때 제거가 되는 방법은 primary storage pool을 데이터 중복 제거를 위해 준비된 storage pool로 복사하는 방법, primary storage에서 중복 제거가 가능한 다른 primary storage로 이동하는 방법 등이 있다[6].

Tivoli Storage Manager 서버와 백업 저장소인 클라이언트는 암호화된 파일을 중복제거 불가능하다. 만약 암호화된 파일이 데이터 처리 과정동안 발견되면 파일은 중복제거 되지 않고 메시지가 남는다.

중복제거 후 저장 방법은 256-bit Advanced Encryption Standard을 사용하고 암호화, 복호화를 위한 키가 키 매니저를 통해 전달이 된다[5].

2) ZFS

ORACLE사의 ZFS는 블록 단위의 중복 제거를 한다. 이때 ZFS는 압축과 암호화와 중복제거가 함께 일어난다. ZFS는 checksum 함수가 암호학적으로 강하므로 storage pool에서 모든 블록의 유일한 블록을 제공하고 256bit 블록의 checksum을 이용하여 중복 제거한다[14].

이 때 checksum은 fletcher4를 이용한다. 이것은 임의 랜덤 해쉬 함수가 아니기 때문에 충돌가능성이 있다. 또한 그것은 확인 옵션과 함께 결합될 때만 적합하다[13]. 확인 옵션은 들어오는 데이터 블록 전체를 확인한다.

데이터는 128,192,256bit 길이의 key와 CCM또는 GCM 운용 모드로 암호화된다[15]. 이때, 암호화된 데이터를 중복제거 하기 위해서는 CCM만 사용가능하다.

key는 128, 192, 256 bit의 AES를 사용하여 키들을 암호화한다[14].

3) Acronis Backup

Acronis사의 Acrnois Backup에서 실행하는 중복제거는 source에서 일어나기도하고 target에서 일어나기도 한다. 이때 fingerprint 혹은 checksum을 각 데이터 블록마다 사용하며 이것은 hash value라고 부른다.

deduplication at source는 블록단위 백업에서 4KB마다, 파일 단위 백업에서는 256KB이하의 데이터 블록으로 중복 제거를 실행한다. 데이터 블록을 저장소에 보내기 전에 agent는 storage node에다가 블록의 hash value이 이미 존재하는지 묻는다. 만약 그렇다면 hash value만을 보내고 그렇지 않다면 데이터 블록을 보낸다.

deduplication at target은 저장소에 백업을 완료한 후에 데이터 블록들은 임시 파일에서 중복제거 데이터 저장공간으로 이동한다. hash value와 링크가 저장된다. 모든 데이터 블록을 모은 후에 임시 파일이 지워진다. 결과적으로 데이터 저장소는 유일한 데이터 블록만을 가지게 된다.

암호화되거나 표준 사이즈로 잘리지 않은 블록은 중복제거가 불가능하다. 중복 제거가 불가능 한 것은 데이터를 아무것도 계산하지 않고 그대로 전달한다[10].

중복 제거 후 CBC모드를 사용한 Advanced Encryption Standard(AES)로 암호화되어 저장된다. 암호화시에 키는 사용자가 지정한 키의 크기(128, 192, 256bit)에 따라 랜덤으로 생성한다. 이 키는 다시 AES로 암호화되는데 이때 키는 사용자 비밀번호를 기반으로 SHA256을 이용한 해쉬 값으로 만들어진다.[11]

4) Simpana

Commvault사의 Simpana는 데이터 중복 제거를 위해 데이터 블록을 위한 signatures를 생성한다. 데이터 블록의 크기는 128KB로, source에서 읽고 해쉬 알고리즘을 이용하여 데이터 블록을 위해 유일한 signature를 생성한다. 데이터 블록은 압축되고, 선택적으로 암호화 가능하다. 암호화된 데이터를 전송하는 경우는 복호화 후에 중복제거를 한다. 그 새로운 signature은 기존의 signature들을 포함하는 Deduplication Database(DDB)에 데이터 블록이 존재하는지 비교한다. 만일, signature가 존재한다면 DDB는 목적지 저장소에서 한 번 더 사용되었다고 기록하고 중복된 데이터를 지운다. 만약에 중복되지 않았다면 그 새로운

signature은 DDB에서 더해지고 그 연관된 MediaAgent는 목적지 저장소에다가 데이터 블럭과 링크 정보를 모두 쓴다[12].

전송시 혹은 저장시에는 CBC를 이용한 256bit-Advanced Encryption Standard를 사용하여 암호화 한다. 이때 초기화 벡터와 키는 의사 난수 생성기인 ANSI 9.31을 이용하여 생성된다. 그리고 키는 다시 RSA로 암호화 된다.

4. 결론 및 향후 전망

위의 현황 이외에도 Dell사의 DR series, Netapp의 Ontap9등 많은 서비스에서 서버로 보내는 과정에서 암호화된 데이터가 아닌 평문 데이터를 보낸다. 그러나 통신 프로토콜의 보안만 믿고 데이터를 전송하는 것은 안전하다고 볼 수 없다. 보내는 데이터는 암호화된 데이터를 사용해야한다.

이때, 암호화한 데이터로 중복제거하기 위해 같은 키를 써야 한다. 그러나 모든 사람이 같은 키를 쓰는 것은 위험하다. 데이터 블록마다 같은 키를 쓰게 해주는 Convergent Encryption(CE)이 암호화된 데이터를 전송 및 비교하여 데이터 중복 제거를 가능하게 해준다. 그러나 CE는 평문을 해쉬한 값을 키로 사용하기 때문에 전수조사에 약하다는 문제가 있다.

이 문제를 해결하기 위해 키 관리에 대한 연구가 진행될 것으로 예상된다. 위의 서비스 4개의 경우 키가 단순하게 암호화 되고 있다. 위와 같은 단순한 암호화가 아닌 여러 겹으로 암호화하는 방법으로 진행될 것으로 예상된다.

또한 소유권 증명에 관한 방법도 중요한 요소 중 하나로 소유권 증명이 되지 않으면 데이터와 키를 전달하는 것이 불가능하다. 따라서 소유권 증명에 관한 방법도 발전할 것으로 예상된다. 소유권 증명은 서버가 할 수도 있고 제 3의 신뢰기관이 할 수도 있다.

참고문헌

[1] Qinlu He, Zhanhuai Li, Xiao Zhang “Data Deduplication Techniques” ,2010,

[2] Chun-I Fan, Shi-Yuan Huang, Wen-Che Hsu, “Encryted Data Deduplication in Cloud Storage” ,2015,10th Asia Joint Conference on Information Security

[3] Kyungsu Park, Ji Eun Eom, Jeongsu Park, Dong Hoon Lee, “Secure and Efficient Client-side Deduplication for Cloud Storage” , ,Journal of The Korean Institute of Information Security & Cryptology Vol.25, No.1, Feb, 2015

[4] http://www.ibm.com/support/knowledgecenter/SS8TDQ_6.4.0/com.ibm.itsm.client.doc/c_dedup.html

[5] http://www-01.ibm.com/support/docview.wss?uid=swg27009625

[6] https://www.ibm.com/support/knowledgecenter/SSSR2R_7.1.0/com.ibm.itsm.srv.doc/c_dedup_srv_ovw.html

[7] https://software.dell.com/docs/dell-dr-series-purpose-built-deduplication-appliances-technical-brief-19576.pdf

[8] ttps://software.dell.com/documents/dr-series-backup-and-deduplication-appliances-datasheet-68537.pdf

[9] https://software.dell.com/docs/how-dell-appassure-global-deduplication-minimizes-your-disk-storage-needs-technicalbrief-24345.pdf s

[10] https://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwi-p7PP247PAhUI5mMKHZrhC8MQFggdMAA&url=http%3A%2F%2Fwww.acronis.com%2Fen-us%2Fdownload%2Fdocs%2FABD%2Ftechnical-whitepaper%2F&usg=AFQjCNGoQgFzqwFhOjpNn_-yYa0ATp0ZOQ&sig2=jZlD83oSFv4MxeilbsljCg&bvm=bv.132479545,d.dGo

[11] http://www.acronis.com/en-us/support/documentation/AcronisBackup_11.7/#546.html

[12] https://documentation.commvault.com/commvault/v10/article?p=features/deduplication/c_deduplication_overview.htm#Source_Side__Client_Side__Deduplication

[13] https://blogs.oracle.com/bonwick/entry/zfs_dedup

[14] http://www.oracle.com/technetwork/articles/servers-storage-admin/manage-zfs-encryption-1715034.html

[15] http://docs.oracle.com/cd/E36784_01/html/E39134/zfsencrypt-1.html

[16] John R.Douceur, Atul Adya, William J.Bolosky, Dan Simon, Marvin Theimer, “Reclaiming Space from Duplicat Files in a Serverless Distributed File System” , Proceedings, 22nd International Conference on IEEE. PP.617-624, 2002,

[17] Mihir Bellare, Sriram Keelveedhi, Thomas Ristenpart, “DupLESS: Server-Aided Encryption for Deduplicated Storage” , Proc. of the 22nd USENIX conference on security, pp.179-194, 2013

[17] Mihir Bellare, Sriram Keelveedhi, Thomas Ristenpart, “DupLESS: Server-Aided Encryption for Deduplicated Storage” , Proc. of the 22nd USENIX conference on security, pp.179-194, 2013

[18] Hyun-il Kim, Cheolhee Park, Dowon Hong, Changho Seo, “Encrypted Data Deduplication Using Key Issuing Server”, Journal of KIISE, vol.43, No.2 , pp143-151,2016

[20] Cheolhee Park, Dowon Hong, Chanho Seo, Ku-Young Chang, “Privacy Preserving Sourc Based Deduplication In Cloud Storage”, Journal of Korea Institue of Information Security & Cryptology, VOL.25, NO.1, Feb, 2015

[21] Mihir Bellare, Keeveedhi, Thomas Ristenpart “Message Locked Encryption and Secrue Deduplication”, proceedings of Eurocrpt, March, 2013.
[22] Won-Bin Kim, Im-Yeong Lee “Secure Data-deduplication Scheme for protect of Data Ownership Using Bloom Filter”, 2016