Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.6, Dec. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.6.1293

금융기관의 테스트시스템 데이터 보안통제 모델 연구

최 영 진,† 김 정 환, 이 경 호‡
고려대학교 정보보호대학원

A Study on Data Security Control Model of the Test System in Financial Institutions

Yeong-jin Choi,† Jeong-hwan Kim, Kyeong-ho Lee‡
Korea University, Graduate School of Information Security

요 약

2014년 카드사 개인정보유출 사고의 원인은 테스트시스템에서 원본 데이터가 사용되었기 때문이다. 금융감독원 전자금융감독규정과 금융회사 정보기술(IT)부문의 정보보호업무 모범규준에는 테스트시스템에서 고객을 식별하는 정보는 변환하여 사용하도록 규정하고 있다. 금융회사는 이 지침에 따라 고객식별정보를 변환한 데이터를 테스트시스템에 적재하여 사용한다. 하지만, 테스트 과정에서의 사용자 실수 또는 기술적, 관리적 보안의 미비 등으로 의도치 않게 실제 개인식별정보가 유입될 수 있으나, 이를 통제 및 관리하는 프로세스는 현재 연구된 바가 없고, 감독기관의 컴플라이언스 위반 가능성을 높이는 원인이 되고 있다. 본 논문에서는 이러한 테스트시스템의 변환 미확인 고객식별정보를 관리 및 통제함으로써 감독기관의 컴플라이언스 위반 가능성을 없애는 프로세스를 제시 및 실증하고, 그 효과성을 확인해 본다.

ABSTRACT

The cause of privacy extrusion in credit card company at 2014 is usage of the original data in test system. By Electronic banking supervision regulations of the Financial Supervisory Service and Information Security business best practices of Finance information technology (IT) sector, the data to identify the customer in the test system should be used to convert. Following this guidelines, Financial firms use converted customer identificaion data by loading in test system. However, there is some risks that may be introduced unintentionally by user mistake or lack of administrative or technical security in the process of testing. also control and risk management processes for those risks did not studied. These situations are conducive to increasing the compliance violation possibility of supervisory institution. So in this paper, we present and prove the process to eliminate the compliance violation possibility of supervisory institution by controlling and managing the unidentified conversion customer identification data and check the effectiveness of the process.

Keywords: Personally Identifiable Information, Unidentified Conversion Data, Compliance, Data Security Control Model

I. 서 론

금융회사는 정보유출을 차단하기 위해서 외부 및 내부의 물리적, 관리적 보안을 통해 정보를 관리 하고 있다. 하지만 이런 노력에도 불구하고 정보 유출 사고는 지속적으로 발생한다. 금융회사는 한번 발생된 보안 사고에도 큰 대가를 치르게 된다. 2014년 발생한 카드3사의 정보유출 사고로 인한 카드 해지와 재발급, 창구방문, 업무폭증 및 영업 손실 등의 총 예상손실을 최소 1천억 원 이상으로 추정하고 있다[1].

접수일(2014년 10월 2일), 수정일(2014년 11월 10일), 게재확정일(2014년 11월 12일)
† 주저자, imstart@korea.ac.kr
‡ 교신저자, kevinlee@korea.ac.kr(Corresponding author)

이 사건의 원인은 예상치 못하게 테스트시스템에서 실제 고객식별정보가 포함된 원본 데이터를 사용함으로써 통제수준이 높은 운영시스템보다 손쉽게 정보를 갈취할 수 있었기 때문이며, 이것은 테스트시스템에서 실제 데이터를 사용함으로써 운영시스템과유사한 환경에서의 테스트를 통해 테스트 결과의 완성도를 높인다는 생각에 비롯되었다. 하지만 금융감독원 전자금융감독규정 제13조(전산자료 보호대책) 10항에는 '이용자 정보의 조회·출력에 대한 통제를 하고 테스트 시 이용자 정보사용 금지 (다만, 부하 테스트 등 사용이 불가피한 경우 이용자 정보를 변환하여 사용하고 테스트 종료 즉시 삭제하여야 한다) 〈개정 2013.12.3.〉 하여야 한다.'고 정하고 있다[2].

그리고 개인정보보호법 제4장 개인정보의 안전한 관리 제29조(안전조치의무)는 '개인정보처리자는 개인정보가 분실·도난·유출·변조 또는 훼손되지 아니하도록 내부 관리계획 수립, 접속기록 보관 등 대통령령으로 정하는 바에 따라 안전성 확보에 필요한 기술적·관리적 및 물리적 조치를 하여야 한다.'고 규정하고 있다[3].

본 논문에서는 위의 규정과 법에 나오는 '이용자정보', '개인정보 용어'를 통일하여 개인식별정보(Personally Identifiable Information: PII)로 표기한다.

국내 금융회사들은 이러한 금융감독원의 규정에 따라 테스트시스템에서 개인식별정보를 가진 DB(Data Base) 테이블을 변환하고 적재하는 프로세스를 운영하고 있다. 그러나 변환적재 이후에 프로그램 개발 및 유지보수에 의해 발생하는 미변환 개인식별정보 항목은 감독기관의 컴플라이언스 위반 가능성을 높이는 취약점으로 작용하고 있다.

따라서 본 논문에서는 금융회사에서 운영되고 있는 테스트시스템의 데이터 변환 적재 프로세스를 파악해보고, 변환 적재 단계 이후에 유입되는 개인식별정보의 변환 미확인 데이터를 관리하고 통제함으로써 금융감독원의 컴플라이언스 위반 가능성을 없애는 효율적 관리 및 통제 프로세스를 제시하고 테스트를 통해 이를 입증해본다.

II. 금융회사 테스트시스템 운영 현황

2.1 금융회사의 테스트시스템 변환적재 운영현황

금융회사의 비즈니스는 점점 복잡해지고 있어 실제 운영상황과 유사한 환경에서 테스트가 필요하며, 이러한 테스트의 오류율을 낮추기 위해서는 테스트시스템에 충분한 데이터를 확보하여 일정기간동안 유지해야만 양질의 테스트결과를 얻을 수 있다. 따라서 금융회사들의 Table 1.에서처럼 정기적인 개인식별정보의 변환 및 데이터 적재 프로세스를 운영하고, 필요에 따라 수시 변환 적재가 가능하도록 하고 있다. 적재시간은 적재용량에 따라 절대적으로 비례하지는 않는데, 이것은 각 회사마다 개인식별정보 변환 및 적재작업에 투입하는 CPU, 스토리지 등의 자원사용수준이 다르기 때문이다.

Table 1.에서 조사된 바와 같이, 정기적인 적재 프로세스에서는 개인식별정보는 모두 변환되어 적재된다. 또한, 각 회사들이 운영하고 있는 변환 프로세스를 통해 통제되는 적재작업인 경우, 개인식별정보가 미변환되어 테스트시스템에 유입되지 않는다. 그러나 이러한 통제절차를 거친 적재 이후 테스트과정에서 개인식별정보의 변환 미확인 데이터를 사용한 사용자의 실수 또는 의도적인 적재시도 및 기술적 보안의 미비 등으로 규정에 위반되는 개인식별정보 변환 미확인 데이터가 불가피하게 테스트시스템에 유입되는 경우가 발생할 수 있다. 이는 기존의 정형화된 적재 프로세스에서 관리, 통제되지 않으므로 별도의 프로세스가 필요하다. 테스트시스템에 이렇게 의도치 않게 유입되는 변환 미확인 데이터의 유입경로를 분기 기준의 정기 적재주기를 적용하고 있는 A금융사의 사례로 확인해보려고 한다.

Table 1. Conversion and load process of PII in some financial institutions

Co., Ltd.	Load Vol.	Load Cycle	Elapsed Time (hours)	Nonscheduled Load
A	15 TB	Quarter	28	Yes
B	20 TB	Quarter	N/A	Yes
C	4 TB	Quarter	N/A	Yes
D	7 TB	Quarter	70	Yes
E	4 TB	Quarter	40	Yes

2.2 A금융사의 테스트시스템 변환 및 적재 프로세스

A금융사는 국내 최대용량의 z/OS[1] 기반의 메인프레임을 운영하고 있는 금융회사이다. A금융사는 DB2[2] DBMS를 사용하며, 정기적 적재 처리 프로세스로

1) z/OS: IBM에서 개발한 메인프레임 컴퓨터용 64비트운영 체제[4].
2) DB2: IBM이 개발한 RDB를 지원하는 데이터베이스

운영시스템의 DB 테이블 백업 데이터를 파일로 생성하고, 개인식별정보를 변환한 후, 테스트시스템에 적재하는 방식으로 처리한다. 또한, 수시 적재 시 에도 운영시스템에서 테스트시스템으로의 우회 적재를 통제하고 있다.

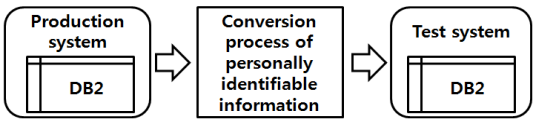

Fig. 1. General conversion and load process by case study of financial institution 'A'

A금융사는 Fig.1.과 같이 운영시스템의 실제 개인식별정보를 안전하게 변환 후, 테스트시스템에 적재하여 사용하고 있다.

2.3 변환 미확인 데이터의 발생경로 사례 분석

A금융사의 정기적재 이후 다음 도래하는 적재시점까지 개인식별정보는 변환하여 테스트를 수행하게 되어 있다. 하지만 Fig.1.에서 테스트시스템에 정상 변환 데이터가 적재된 이후 Fig.2.에서처럼 배치작업에 의한 수시적재, 각종 채널을 이용한 테스트 거래, 테스트를 위해 외부 기관으로부터 입수한 데이터의 적재작업 등을 수행하면서 변환여부를 미처 확인 못한 데이터가 개인식별정보 변환 대상 테이블에 반영되어 로우(row)건수를 증가시킬 수 있다.

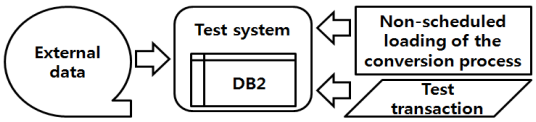

Fig. 2. Data inflow to test system not controlled by assigned conversion process

A금융사는 분기 단위로 정기적 적재를 수행하고 있기 때문에, 약 3개월간의 별도 경로를 통한 데이터 유입 추이의 확인이 필요하며, 이러한 별도 경로를 통한 데이터도 반드시 정규화된 변환 프로세스를 사용하도록 되어 있어 규정을 준수하고 있다. 하지만, 각 회사의 기술적 보안, 관리적 보안의 미비 등으로 정규화된 변환 프로세스를 통하지 않고 테스트를 진행한 개발자의 의도하지 않은 실수, 변환 여부를 검증하지 않은 외부데이터의 유입 등으로 인해 실제 개인식별정보가 포함된 데이터가 테스트시스템에 유입되는 경우 금융감독원의 컴플라이언스를 위반하게 되는 원인이 된다. 이러한 개인식별정보를 포함한 데이터가 해당 칼럼의 변환여부가 미확인되어 테스트시스템의 DB 테이블에 유입된 선별적 검증이 필요한 데이터를 '변환 미확인 데이터(Unidentified Conversion Data: UCD)'라 정의한다.

2.4 변환 미확인 데이터의 발생유형 분석

테스트시스템의 정기적 적재 이후에 발생하는 변환 미확인 데이터의 발생유형을 DBMS log[3]의 발생기준에 따라 다음과 같이 분석하였다.

- DBMS log 발생: 테스트 온라인거래, 배치작업 등 어플리케이션 기반 테스트를 통한 데이터의 삽입, 갱신, 삭제 발생
- DBMS log 미발생: DBMS log를 생성하지 않도록 한 솔루션을 이용한 데이터 적재작업 등의 데이터 기반 테스트를 통한 데이터의 삽입, 갱신, 삭제 발생

III. 컴플라이언스 위반에 대한 취약성 분석

Ⅲ에서는 감독기관의 컴플라이언스 위반 가능성을 높이는 취약성을 없애는 프로세스를 도출하기 위해 Ⅱ에서 알아본 테스트시스템 적재 프로세스 기반에서 영향변수를 도출하고 영향수준을 분석하여 통제 가능한 영향변수를 확인해보고, 이에 따른 대책(countermeasure)을 제시해본다.

3.1 영향변수의 도출 및 영향수준 정의

3.1.1 영향변수의 도출

감독기관의 컴플라이언스 위반 가능성을 높이는 취약성은 테스트시스템에 정규화된 변환 프로세스를 통하지 않고 발생된 데이터의 노출수준에 비례한다.

서버 상품 패키지로 IBM의 주요 OS에 특화되었으나, 1990년대 이후 타 플랫폼도 지원하도록 변화함[5].

3) DBMS log: 삭제, 수정, 생성 등 데이터베이스의 모든 활동에 대한 흔적을 남긴 특별한 테이블로, 데이터 복구 또는 관리 목적으로 사용됨[6].

즉, 해당 데이터의 노출량이 많을수록, 노출주기가 길어질수록 취약성은 높아진다. 따라서 영향변수를 아래와 같이 도출하였다.

- 데이터 적재주기 : 데이터 적재주기가 길수록 노출주기가 길어져 취약성은 높아진다.
- 변환 미확인 데이터 발생건수 : 변환대상 DB 테이블에 개인식별정보의 추가 및 변경을 주는 변환 미확인 데이터 발생건수가 많을수록 노출량이 많아져 취약성은 높아진다.

3.1.2 컴플라이언스 위반에 대한 취약수준 정의

정기적 적재 이후 다음 도래하는 정기적 적재시점까지 A금융사의 변환 미확인 데이터 증가 추이 확인을 위해 일 단위 증가량을 산출하여 주 단위로 환산한 누적흐름은 Fig.3.과 같다.

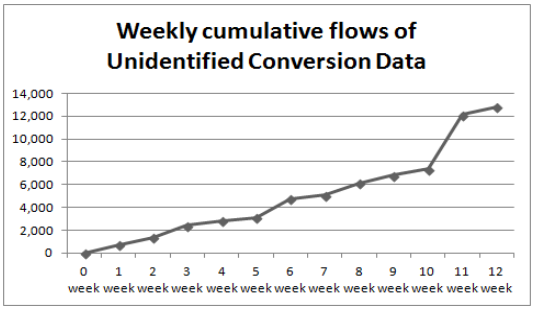

Fig. 3. Cumulative flows by analyzing weekly range data of Unidentified Conversion Data

Fig.3.에서 주 단위, 일 단위 등 추출구간의 기준에 따라 특정구간의 기울기가 커지며, 누적현황이 타 구간보다 급하게 증가하는 경우는 대량 프로그램의 변경에 따른 테스트가 많거나, 별도의 비정형 데이터 적재가 발생하는 등의 예외적인 경우이다.

실례로 Fig.3.에서 변환 미확인 데이터의 유입량이 급격히 증가한 흐름을 보인 10주~11주 사이에는 규모가 매우 큰 개발 프로젝트의 프로그램 적용 일정이 집중되었던 시기인 것으로 확인되었고, 해당 프로젝트를 수행하면서 발생한 데이터는 모두 변환된 데이터였음을 확인하였다.

따라서 해당 구간에 대해 변환된 데이터로 확인된 로우건수를 제외한 후, 보정된 변환 미확인 데이터의 누적현황은 Fig.4.와 같다.

Fig.4.에서처럼 실제 데이터를 통해 확인한 변환 미확인 데이터 증가수준이 다른 구간에 비해 급격한 변화를 보이는 경우, 해당 구간에서 발생한 변환 미확인 데이터의 발생경로 및 변환여부에 대한 확인이 필요하다. 이러한 확인 및 보정작업을 통해 얻은 결과는 Fig.4.에서와 같이 일반적으로 정기적 적재이후 추출구간별로 일정 수준의 추가유입량이 증가하는 직선 그래프에 수렴한다. 따라서 추출구간의 평균유입량을 절대적으로 확정하기는 어렵지만, 각 회사가 유의미한 수준의 기간에 대한 유입량을 확인함으로써 직선 그래프에 수렴하는 기준추출구간의 평균 유입량을 아래 제시하는 공식에 따라 대입하여 계산하면, 정기적 적재 이후 다음 도래하는 정기적 적재시점까지의 변환 미확인 데이터 노출수준을 확인할 수 있다.

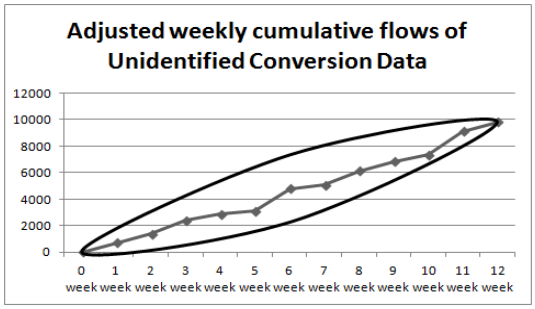

Fig. 4. Adjusted cumulative flows by analyzing weekly range data of Unidentified Conversion Data

즉, 3.1.1에서 도출한 영향변수와 취약성의 관계에 의해 컴플라이언스 위반에 대한 취약수준을 변환 미확인 데이터의 노출수준으로 정의하여 아래와 같이 일반화한 공식에 의해 산출한다.

- x : 일일 변환 미확인 데이터 발생 로우건수
- n : 현재 변환주기(일)
- k : 현재 변환주기 내 순차적 일수

변환 미확인 데이터 노출수준 산출 공식

$$UCD = \sum_{k=0}^{n-1} x(k+1)$$

위의 공식에서, n은 현재 변환주기를 나타내며, 2.1에서와 같이 3개월 단위로 정기적 적재를 수행하는 경우, n=90이 된다. k는 n값 이내의 순차적 일수

를 의미하며, n=90인 경우, k=0부터 k=89까지의 값을 대입한다. k=0은 최초기준일로, 정기적 적재작업이 완료된 익일 0시부터 익익일의 0시 이전까지를 의미한다. 최초기준일을 k=1로 정의하지 않은 것은 3.2의 대책(countermeasure)을 제시한 공식에서 변환 미확인 데이터의 재변환주기를 적용함에 있어, k=1로 적용 시 재변환주기 발생지점에서 공식에 추가 보정이 필요하기 때문이며, 공식을 간략히 정리하기 위해 k=0으로 정의하였다.

A금융사의 정기적 적재 주기는 3개월이고 테스트시스템에 유입되는 변환 미확인 데이터의 일평균 유입량을 일 100,000건으로 가정하면, 아래 Fig.5.에서와 같이 변환 미확인 데이터의 노출수준을 확인할 수 있다.

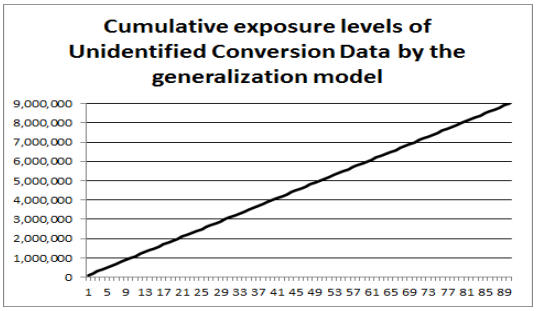

Fig. 5. Cumulative exposure levels of Unidentified-Conversion Data by the generalization model proposed in this paper

이를 공식에 적용하여 x=100,000, n=90, k=0~89를 대입하면, 변환 미확인 데이터의 노출수준은 409,500,000이 되며, 이 수치는 직선그래프 범위내의 면적을 의미한다. 이렇게 변환 미확인 데이터가 테스트시스템에 계속 상존하는 경우, 다음 도래하는 정기적 적재시점까지 노출수준은 지속적으로 증가하므로 이를 최소화하기 위한 대책이 필요하다.

3.2 대책(Countermeasure) 제시

3.1에서 도출한 컴플라이언스 위반에 대한 취약수준을 낮추기 위해 각 영향변수의 통제 가능 수준을 파악해 본 결과는 다음과 같다.

- 데이터 적재주기 : 데이터 적재주기를 줄이면 컴플라이언스 위반 수준을 높이는 취약성을 낮출 수 있지만, 2.1에서 확인한 바와 같이 정기적 적재는 수일이 걸리는 작업으로 해당기간 동안 테스트시스템의 프로그램 개발자 테스트가 불가한 제약사항이 발생하고, 이는 실무 영향도가 매우 크므로 통제 불가함.
- 변환 미확인 데이터 발생건수 : 정기적 적재 이후 테스트 등에 따라 불가피하게 발생하므로 통제 불가함.

위의 결과와 같이, 기존 영향변수를 통제함으로써 컴플라이언스 위반 수준을 높이는 취약성을 낮추는 것은 불가능하다. 따라서 불가피하게 발생하는 변환 미확인 데이터를 빠르게 추출 및 재 변환 처리하여 해당 데이터의 노출주기를 줄이는 것이 필요하며, 본 논문에서는 이를 통제하기 위한 방안을 제시하고 실증을 진행하였다. 이를 반영하면 컴플라이언스 위반에 대한 취약수준은 아래와 같이 재정의 할 수 있다. 참고로, 재변환주기를 일 단위로 설정한 것은 영업시간 중 재변환 처리 시 프로그램 개발자 테스트 제약사항 등이 발생하므로 최소 일 단위의 재변환 처리가 유효하다고 판단하였다.

이는 각 회사의 자원투입 가능량, 전산 프로세스 등을 감안하여 적절한 가이드라인을 설정하여 적용하는 것이 바람직하며, 제시된 일 단위의 재변환 주기를 취약수준을 더욱 감소시키기 위해 시간단위로 적용하는 것이 해당 회사의 자원상황과 전산 프로세스에 영향도가 없다면 시간단위의 적용도 고려해 볼 수 있다. 또한, 이러한 프로세스 적용을 통해 일정량 이상의 테스트시스템의 변환 미확인 데이터 유입이 인지되는 경우, 회사의 컴플라이언스 담당자에 알리는 관리적 보안 프로세스를 추가함으로써 컴플라이언스 위반에 대한 취약수준을 낮출 수 있다.

- x : 일일 변환 미확인 데이터 발생 로우건수
- n : 현재 변환주기(일)
- k : 현재 변환주기 내 순차적 일수
- c : 변환 미확인 데이터 재변환 주기

재변환 적재주기를 적용한 노출수준 산출 공식

$$UCD^{*} = \sum_{k=0}^{n-1} x\,(k \bmod c + 1)$$

아래 Fig.6.는 Fig.5.에서 지속적으로 증가한 변환 미확인 데이터의 노출수준을 제시한 공식에 따라 재변환 적재주기인 c값을 90일, 30일, 7일, 1일로 적

용했을 때, 노출수준의 감소추이를 보여준다.

아래 Fig.7.은 Fig.6.의 변환 미확인 데이터 노출수준의 변화에 따른 총노출량을 토대로 재변환 적재주기의 적용수준에 따른 총노출량의 감소비율을 나타내었다. 즉, 현재의 재변환 적재주기가 적용되지 않은 상태를 100%로 볼 때, 재변환 적재주기를 빠르게 할수록 변환 미확인 데이터의 노출수준은 재변환 적재주기의 단축수준에 비해 더욱 낮아져 컴플라이언스 위반의 취약수준을 크게 낮춘다.

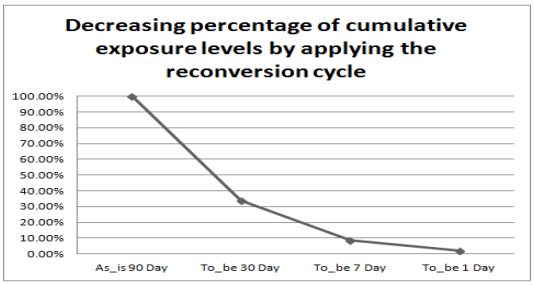

Fig. 6. Cumulative exposure levels of Unidentified-conversion data by the generalization model applied some cases of reconversion cycle

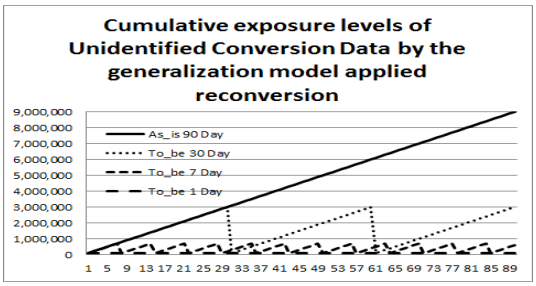

Fig. 7. Decreasing percentage of cumulative exposure levels of Unidentified-conversion data by the generalization model applied the reconversion cycle

Ⅳ에서는 이러한 취약수준을 낮추기 위한 테스트시스템의 변환 미확인 데이터 관리 및 통제 방안을 제시하고, 테스트를 통해 실증해본다.

Ⅳ. 데이터 보안통제 프로세스 테스트

4.1 방안 제시 및 프로세스의 구성

테스트시스템의 변환 미확인 데이터의 검증 및 재변환 적재를 위해 Fig.8.과 같은 방안을 제시한다.

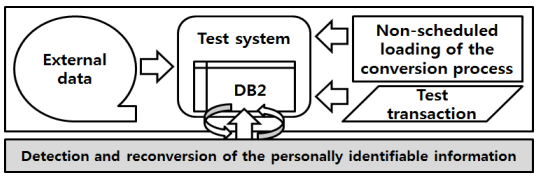

Fig. 8. Methodology for reconversion and reload of Unidentified Conversion Data

Fig.8.은 변환 미확인 데이터가 지속적으로 유입되는 경우의 노출수준을 기존 통제변수인 데이터 적재주기, 변환 미확인 데이터 발생건수의 통제를 통해 낮출 수 없으므로, 유입되는 변환 미확인 데이터에 대해 새로운 통제변수인 '변환 미확인 데이터 재변환 주기'를 적용하여 해당 주기에 따라 재변환 및 해당 DB 테이블에 재반영 함으로써 테스트시스템에서의 노출수준을 낮춰야 한다는 것을 의미한다.
변환 미확인 데이터를 재변환 하고 반영하기 위한 제안하는 프로세스는 다음과 같다.

- 변환 미확인 데이터 검증대상 테이블 추출
- 재변환이 필요한 변환 미확인 데이터 추출
- 변환 미확인 데이터 재변환 처리
- 재변환 처리된 데이터 재적재 처리

Fig.9.는 변환 미확인 데이터를 재변환 하고 반영하는 프로세스를 도식화한 것이다.

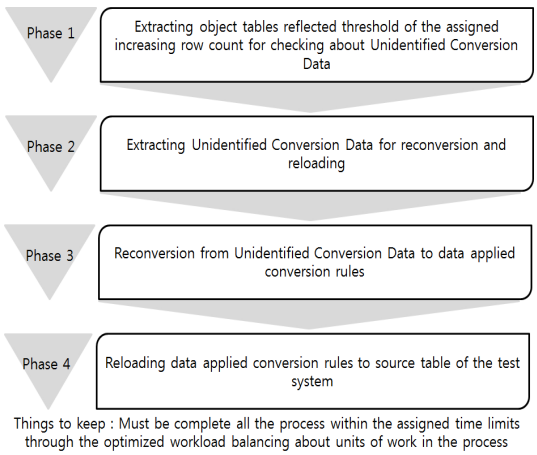

Fig. 9. Process for reconversion and reload of Unidentified Conversion Data

4.2와 4.3에서는 본 논문의 테스트환경인 메인프레임 기반에서 본 프로세스를 효율적으로 구현하기 위한 전산 프로세스를 제시해보고, 4.4에서는 구현된 프로세스의 테스트를 통해 실효성을 검증해본다.

4.2 단계별 전산 프로세스 구현

4.2.1 변환 미확인 데이터가 포함된 대상테이블 추출

4.2.1.1 요구사항 정의

2.1에서와 같이 대용량 데이터베이스를 운영하는 회사에서 변환 대상 전 테이블에 대해 모두 변환 미확인 데이터 발생여부를 검증하는 것은 과도한 자원사용과 시간 제약으로 실효성이 없다.

따라서 변환 미확인 데이터의 재변환 주기를 각 회사의 가이드라인에 맞춰 빠르게 적용하려면 자원사용을 최소화하고 신속하게 변환 미확인 데이터만을 추출할 수 있도록 변환 미확인 데이터 검증이 필요한 대상 테이블을 빠르게 추출하도록 구현해야 한다.

4.2.1.2 데이터 변경사항이 발생한 테이블 추출 방법

4.2.1.1의 요구사항에 따라 변환 미확인 데이터의 검증 및 재변환이 필요한 대상 테이블을 추출하기 위해 정의한 구간에 테이블 데이터의 변경사항이 발생한 변환 대상 테이블을 가장 빠르게 확인하는 방법이 필요하여, 이에 따라 다음의 DB2 시스템 카탈로그(System Catalog)[4] 테이블을 활용하였다.

이 테이블의 정보를 활용하지 않으면 변환 대상 전 테이블에 대해 변환 미확인 데이터의 발생여부를 검증해야 하므로 제한된 시간 내 수행이 불가능하고 과도한 자원사용이 발생한다.

따라서 가장 빠르고 신속한 검증대상 테이블 추출을 위해 사용된 시스템 카탈로그 테이블명은 'SYSIBM.SYSTABLESPACESTATS', 'SYSIBM.SYSTABLES'이며, 이들은 각각 정기적 적재이후 INSERT, UPDATE, DELETE가 발생한 스냅샷(snapshot)[5] 정보와 DBMS에 종속된 논리적 테이블의 정보를 보유하고 있다.

4.2.1.3 임계치 기준의 설정

변환 미확인 데이터가 유입된 변환 대상 테이블의 검증대상 유입수준의 임계치를 각 회사가 유연하게 설정하여 실효성 있는 검증을 수행하도록 하는 것이 바람직하다. 단, 변환 미확인 데이터가 유입된 변환 대상 테이블이 임계치 기준에 충족하지 못해 검증 대상으로 추출되지 않는 경우 관련 테이블의 개인식별정보만 재변환하여 적재하면, 테이블간 RI (Referential Integrity)[6]가 파괴될 수 있으므로 이 부분은 임계치 기준을 설정하는 프로세스 운영 시 각 회사의 정책에 맞게 RI를 확보되도록 구현해야 한다.

예를 들면, DBMS 레벨의 Referential Constraints[7]가 적용되어 있는 경우는 RI가 파괴되는 상황에서 해당 데이터를 재적재시 오류가 발생하므로 재변환에 따른 RI의 파괴를 적재시점에 인지할 수 있으나, 복잡한 대량 DB 테이블 관리 등의 문제로 테이블의 관리정책을 Referential Constraints를 설정하지 않고, ERD[8] 등의 논리적인 기준으로만 RI를 운영하는 경우에는 임계치 설정 시 재변환에 따른 RI가 파괴된 데이터가 적재시점에 오류 처리되지 않고 적재될 수 있다. 따라서 각 회사가 이 부분을 유념하여 임계치를 설정하여야 하며, 본 논문에서는 이 부분에 대해서 별도로 추가 언급하지 않는다.

4.2.1.4 대상 테이블 추출 알고리즘

자원사용을 최소화한 신속한 변환 미확인 데이터 검증대상 테이블 목록정보를 추출하기 위한 알고리즘을 위의 Fig.10.와 같이 제시한다.

해당 테이블 정보를 추출하는 단계는 크게 2단계로 구성된다. 1단계는 DBMS에 종속된 전체 업무테이블 중 변환대상 테이블 정보와 해당 테이블에 대해 정기적 적재이후 'CUD'[9] 구문이 수행된 현황의 스냅샷

4) System Catalog: 데이터베이스에 관한 중요 정보를 포함한 테이블과 뷰(view)의 집합[7].

5) 스냅샷(snapshot): IT용어로 특정 시점의 상태를 표시한 것

6) RI(Referential Integrity): 관계형 데이터베이스 모델에서 2개의 관련있는 관계 변수 또는 테이블 간의 일관성으로 데이터 무결성을 의미함[8].

7) Referential Constraints: 관계형 데이터베이스 모델에서 2개의 관련있는 테이블의 모든 로우가 RI가 만족되도록 제약사항을 설정하는 것으로, 테이블 생성 또는 테이블 속성 변경시 설정가능함[9].

8) ERD(Entity Relationship Diagrams): 데이터베이스의 논리적 구조를 도식화한 것[10].

9) 'CUD'란, 컴퓨터 소프트웨어가 가지는 기본적인 데이터 처리 기능인 'CRUD'에서 INSERT, UPDATE, DELETE 구문을 통해 DB 테이블에 변경을 발생시키는

정보를 보유한 시스템 카탈로그 테이블을 이용하여, 정기적 적재이후 각 회사가 검증이 필요한 기준을 정한 로우 증가량의 임계치를 적용하여 해당 조건에 부합하는 업무테이블을 추출한다.

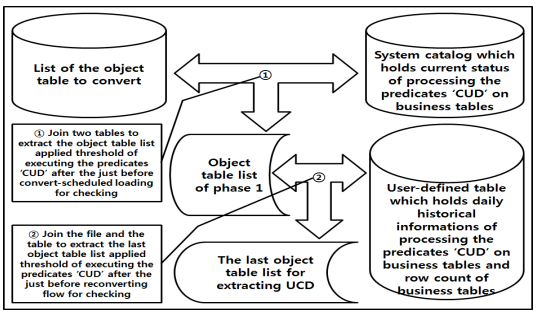

Fig. 10. Algorithm for extracting object tables for checking about Unidentified Conversion Data

2단계는 1단계에서 추출한 업무테이블 목록과 해당 테이블에 대해 테이블의 변동 상황을 구간별로 수집한 히스토리성 정보 테이블을 이용하여, 직전 재변환 및 재적재 처리시점 이후 각 회사가 검증이 필요한 기준을 정한 로우 증가량의 임계치를 적용하여 해당 조건에 부합하는 최종 검증대상 업무테이블을 추출한다. 본 논문에서 사용된 히스토리성 정보 테이블은 일일단위로 수집되었다.

4.2.2 재변환이 필요한 변환 미확인 데이터 추출

4.2.2.1 요구사항 정의

4.2.1에서 추출한 변환 미확인 데이터 검증 대상 테이블에 대해 설정한 재변환 주기에 따라 직전 재변환 재적재 작업 이후에 발생한 변환 미확인 데이터를 모두 추출하도록 한다.

4.2.2.2 정상 변환 데이터를 보관한 복제테이블 생성

변환 미확인 데이터를 실효성 있게 추출하기 위해서는 검증 대상 테이블에서 정기적 적재 또는 수시 적재와 같이 정규화 된 변환 프로세스를 통해 변환된 데이터를 제외하여야 한다. 따라서 변환 미확인 데이터에서 정상 변환 데이터가 제외될 수 있도록 정상 변환 데이터의 별도 보관이 필요하며, 변환 미확인 데이터 추출 시 별도 보관된 정상 변환 데이터가 제외되도록 설계하여야 한다.

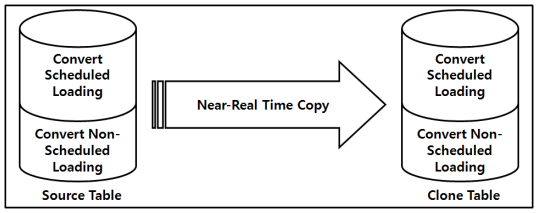

Fig. 11. Methodology for creation and maintenance of clone tables which have only converted data through the assigned conversion process

본 논문에서는 이러한 정규화 된 변환 프로세스를 통한 정상 변환 데이터를 Fig.11.과 같은 방식으로 복제테이블을 생성하여 유지할 것을 제안한다.

해당 복제테이블은 정기적 적재시점에 정기적 적재 데이터로 전체를 모두 재생성한 후, 정규화된 변환 프로세스를 통한 수시적재가 발생하는 경우 준실시간(near real time)으로 해당 데이터를 추가하여 반영하도록 한다.

4.2.2.3 대상 데이터 추출 알고리즘

검증 대상 테이블에서 변환 미확인 데이터를 추출하는 알고리즘을 아래 Fig.12.와 같이 제안한다.

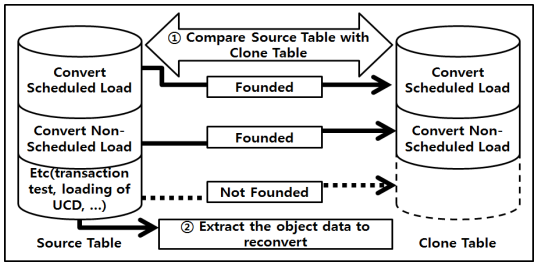

Fig. 12. Methodology for extracting Unidentified Conversion Data for reconversion and reloading

검증 대상 테이블을 Fig.11.에서와 같이 정상 변환 데이터를 준실시간으로 반영하고 있는 복제테이블과 기본키(primary key)[10]를 비교하여 복제테이블에

'CREATE', 'UPDATE', 'DELETE'를 의미한다[11].

10) 기본키(primary key): 관계형 데이터베이스(RDB)에서 관계(데이터베이스 테이블) 내의 특정 열을 일의적으로 식별할 수 있는 키 필드로 주 키(major key)라고도 함. 유니크(unique)한 속성을 가짐[12].

미존재하는 데이터를 확인이 필요한 변환 미확인 데이터로 추출해낸다. 따라서 이러한 방법을 통하면 간단히 변환 미확인 데이터를 추출해낼 수 있다.

4.2.3 변환 미확인 데이터 재변환 및 재적재 처리

4.2.2에서 추출한 변환 미확인 데이터 로우를 정해진 변환룰을 적용하여 재변환후 해당 테이블에 재반영한다. 일반적으로, 소량의 데이터의 경우 재변환 및 재적재 알고리즘을 탑재하여 재변환 및 재적재 프로세스를 동시 처리하도록 하며, 대량 데이터인 경우 메모리 내부를 통한 두 프로세스의 동시처리가 CPU 자원에 부하를 일으킬 수 있으므로, 해당 데이터를 파일로 생성 및 파일변환 후, 테이블에 갱신하도록 한다.

4.3 전산 프로세스 통합 모델

4.2에서 제시한 단계별 프로세스를 통합한 모델은 Fig.13.과 같다.

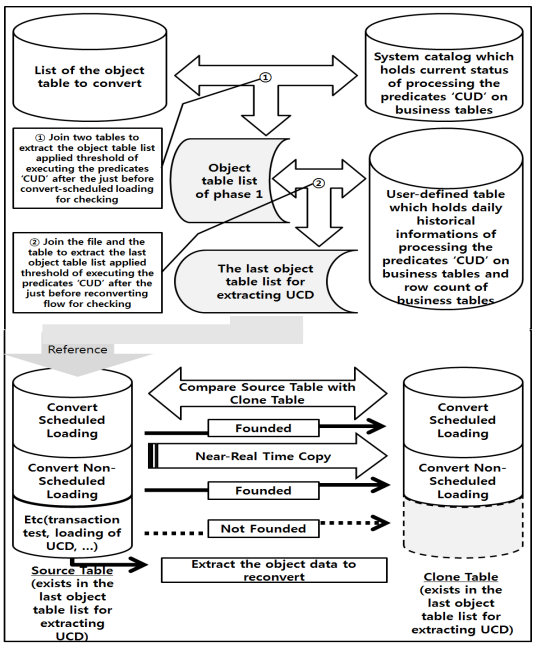

Fig. 13. Integrated process model for extracting Unidentified Conversion Data for reconversion and reloading

요약하자면, 변환 대상 목록 정보 테이블 및 전체 업무테이블에 대한 주기별 로우 변동 상황의 기록을 보관한 별도 테이블을 구성하고 DBMS가 제공하는 시스템 카탈로그 테이블을 활용함으로써, 대용량 데이터를 운영하는 경우에도 매우 빠르게 정기적 적재 이후 추가로 유입된 로우의 임계수준를 반영하여 실효성 있게 변환 미확인 데이터의 검증이 필요한 대상 테이블을 찾아낼 수 있다.

또한, 최종 추출된 검증대상 테이블 정보를 참조하여 테스트시스템의 원본 테이블을 정상 변환 데이터만을 보유한 복제 테이블과 비교함으로써 변환 미확인 데이터만을 추출해 낼 수 있다.

4.4 테스트

4.4.1 변환 미확인 데이터가 포함된 대상테이블 추출

본 논문의 테스트는 다음 환경에서 진행하였다. 테스트를 진행한 시스템은 약 2100 밉스(MIPS) 용량의 CPU를 사용하는 메인프레임 z/OS 운영체제 환경이며, DBMS는 DB2 V10 기반이다. MIPS에 대해서는 V에서 설명하겠다.

먼저, Fig.14.에서와 같이 변환 미확인 데이터 검증 대상 테이블을 추출하기 위한 전체 업무테이블 수는 5,355개이며, 총용량은 약 29.4TB(테라바이트)이다.

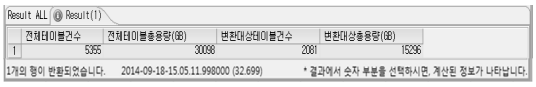

Fig. 14. Confirmation of table count and load volume of object tables which have PII among all the business table

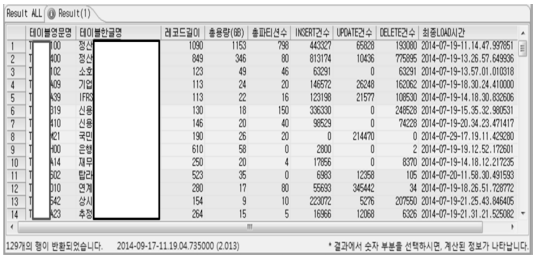

Fig. 15. Result of executing the implemented logic for extracting the object table list applied threshold of executing the predicates 'CUD' after the previous convert-scheduled loading in phase '1'

전체 업무테이블 중 변환 대상 테이블의 총개수 및 용량을 확인해 본 결과, 2,081개 테이블, 약

14.9TB임을 확인하였다. 먼저, 변환 대상 테이블 중에서 정기적 적재 이후 현 시점까지 'CUD' 구문이 수행된 현황의 스냅샷 정보를 토대로 INSERT, UPDATE가 1,000건 이상 발생한 임계치 기준을 설정하여 테스트해 본 결과는 Fig.15.와 같다.

가장 최근의 정기적 적재작업은 2014년 7월 16일에 완료되었음을 확인하여, 정기적 적재작업 직후인 2014년 7월 17일부터의 변경현황을 검증하도록 설정하였다. 그 결과, 설정한 임계치에 부합하는 테이블은 총 129개가 도출되었고, 이 결과를 도출하는 로직(logic)의 수행시간은 약 2초였다.

아래 Fig.16.은 Fig.15.에서 테스트를 수행한 로직을 반영하여 직전 재변환 및 재적재 처리시점 이후 각 회사가 검증이 필요한 기준을 정한 로우 증가량의 임계치를 적용하여 해당 조건에 부합하는 최종 검증대상 업무테이블을 추출한 프로그램 테스트 결과이다.

Fig. 16. Result of executing the implemented program for extracting the last object table list applied threshold of executing the predicates 'CUD' after the previous reconverting and reloading flow in phase '2'

본 테스트에서는 테이블별 일일 로우증가량이 100건이 넘는 경우를 검증대상으로 판단하도록 설정하였고, 그 결과 총 13개 테이블이 최종적으로 변환 미확인 데이터를 확인하는 대상으로 추출되었다.

최종 검증 대상 테이블을 추출하는 프로그램의 수행시간은 약 10분 10초였다. 정리하면, 총 5,355개의 약 29.4TB 에 달하는 업무테이블 중 변환 미확인 데이터 검증 대상 테이블을 최종 추출하는 작업은 본 테스트 환경에서 약 10여분 소요되었다.

4.4.2 재변환이 필요한 변환 미확인 데이터 추출

변환 미확인 데이터를 추출하는 테스트는 Fig.16.의 결과로 도출된 최종 검증 대상 테이블 13개 중 약 4,200만건의 로우를 보유한 테이블을 대상으로 수행하였다. Fig.12.의 알고리즘을 구현하기 위해 SQL에 'LEFT OUTER JOIN'구문을 사용하여 복제테이블과의 기본키 비교를 수행하였다. 그 결과, 109건의 변환 미확인 데이터 로우를 추출하였으며, 해당 테이블은 '주민사업자번호' 칼럼에 변환룰이 적용되도록 설계되었음을 확인하였다. 또한 해당 테이블의 '주민사업자번호' 칼럼은 변환 대상이나, 필수입력 값이 아니어서 Fig.17.의 추출결과와 같이 일부 로우에서만 입력 값이 확인되었다.

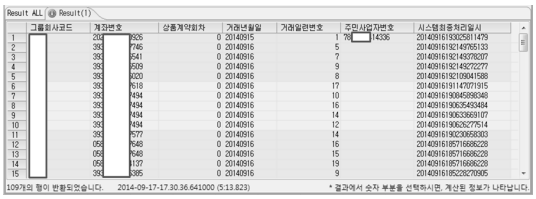

Fig. 17. Result of executing the implemented program for extracting Unidentified Conversion Data

약 4,200만건의 로우를 보유한 테이블에 대해 변환 미확인 데이터를 추출하는 프로그램의 수행시간은 약 5분 13초였으며, 이 중 CPU를 사용한 CPU time은 Fig.18.에서처럼 약 2분 12초로 나타났다.

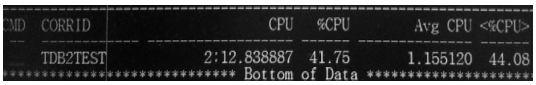

Fig. 18. Result of CPU usage about executing the implemented program for extracting Unidentified Conversion Data

이렇게 산출된 CPU time은 V에서 설명하게 될 워크로드 밸런싱(workload balancing)[11]설정을 위한 자원사용량에 매우 중요한 내용이므로 각 회사들은 회사의 시스템 용량에 대한 전산 단위작업의 CPU time을 책정해 보는 것은 매우 중요하다.

4.4.3 변환 미확인 데이터 재변환 및 반영

Fig.17.에서 추출한 변환 미확인 데이터 중 변환

11) 워크로드 밸런싱(workload balancing): 로드 밸런싱(load balancing)이라고도 하며, CPU, 스토리지, 네트워크 연결, 다중 수행 작업 등의 컴퓨터 자원을 최적으로 분산하는 것을 말함. 이것은 대용량 컴퓨팅 자원을 사용하는 경우, 주어진 자원으로 최적의 성능 및 부하방지를 위해 더욱 중요함[13].

대상 칼럼인 '주민사업자번호'에 입력 값이 있는 로우를 샘플링하여 변환룰을 적용 후 재반영한 결과는 Fig.19.와 같다.

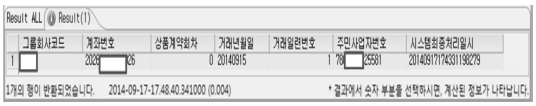

Fig. 19. Result of PII column value after reconversion and reloading

변환 미확인 데이터로 샘플링한 로우의 '주민사업자번호'는 확인 결과 정규화 된 변환 프로세스를 통하지는 않았으나, 사전 변환하여 사용한 칼럼이었으며, 해당 칼럼을 복호화 불가능한 단방향 변환룰에 따라 2차 변환한 값으로 변환된 것을 확인할 수 있었다.

V. 데이터 보안통제 프로세스 효과성 분석

2014년에 관리적 보안의 미비로 인해 발생한 카드3사의 개인정보유출 사고 시 추정한 손실금액을 확인해보고, 본 논문에서 제안한 데이터 보안통제 프로세스의 효과성을 분석해본다.

5.1 자산 가치의 식별

테스트시스템에서 변환 미확인 데이터의 노출은 감독기관의 컴플라이언스 위반의 가능성을 높이고, 권한통제 수준이 운영시스템에 비해 낮은 테스트시스템에 쉽게 접근할 수 있는 경우 카드3사의 개인정보유출 사고와 같이 관리적 보안의 미비에 따른 개인식별정보의 유출가능성을 높인다.

금융감독원과 카드업계에서 추정한 정보유출 카드3사의 정보유출에 대한 3개월 영업정지로 인한 영업손실은 각각 445억원, 289억원, 339억원으로 총 1,072억원에 달하고, 카드재발급 및 우편발송비용 등 후속 처리비용은 A카드사의 추정금액 217억원을 비롯해 총 534억원으로 추정되어, 기업 측면의 실질 손실은 1,606억원으로 추정된다[14].

2014년 3월 금융감독원 전자공시시스템의 사업보고서에 따르면 카드3사의 정보유출 규모는 A카드사 4,300만명, B카드사 2,427만명, C카드사 1,760만명으로 총 8,487만명(중복포함)이며, 김종환 외(2014)는 이러한 정보유출에 따른 정신적 손해를 권홍 외(2012)의 연구결과인 개인들에 대한 설문조사결과 이중양분석택형법을 통한 수용의사금액을 근거로 카드3사의 피해인원을 곱하여 A카드사 30조 140억원, B카드사 62조 9,807억원, C카드사 45조 6,720억원으로 산출하였고, 동일문헌에서 유진호 외(2009)의 연구결과인 손해배상금 산출을 위한 파라미터를 근거로는 A카드사 8조 6,000억원, B카드사 7조 2,810억원, C카드사 5조 2,800억원으로 산출하였다.

김종환 외(2014)는 동일 문헌에서 카드3사가 SK컴즈사건의 실제 소송비율 0.008%를 보수적으로 적용하여 정신적 손해에 대한 소송참여수를 1%를 기준으로 A카드사 860억원, B카드사 485억원, L카드사 352억원을 자체 추정하였음을 말하고 있다[15].

즉, 기존의 연구결과를 근거로 산출한 피해액은 천문학적이나, 카드3사가 자체 추정한 보수적인 기준으로 판단하여도 정신적 손해에 대한 기업측면의 총손실액은 1,697억원에 달한다.

따라서 카드3사의 정보유출로 인한 총손실액은 영업손실금액 및 후속 처리비용, 정신적 손해에 대한 추정손실금액을 보수적으로 합산하여도 3,303억원으로 산출된다.

5.2 대책(countermeasure)에 대한 TCO 산출

본 논문은 테스트시스템의 변환 미확인 데이터의 노출수준을 낮추는 대책 및 전산 프로세스를 제시하여 컴플라이언스 위반 가능성을 높이는 취약성을 없앰으로써, 카드3사의 정보유출사건으로 대표되는 테스트시스템의 데이터 통제 미비로 인한 개인식별정보 유출문제를 해결하도록 제안하고 있다. 이러한 대책 적용에 필요한 전산 프로세스에 대한 TCO(Total Cost of Ownership)을 다음과 같이 산출할 것을 제안한다. TCO란, 회사에서 전산 시스템을 도입할 때 단순히 초기 투자비용만이 아니라 도입 후의 운영이나 유지 보수비용까지 고려하는 것을 의미한다[16].

따라서 데이터 보안통제 전산 프로세스의 초기 투자비용과 유지 보수비용을 고려한 TCO는 다음과 같다.

데이터 보안통제 프로세스의 TCO = 초기 개발비용 + CPU증설비용 + CPU증설에 따른 소프트웨어 사용비용 증가액 + 스토리지 추가 비용

초기 개발비용은 자체인력을 투입, 아웃소싱(outsourcing) 등의 개발방식에 따라 달라지며, 자

체인력 투입하는 경우 기업 측면에서의 실질적 추가 비용은 발생하지 않는다. 유지 보수비용 중 스토리지 추가 비용은 정상 변환 데이터의 복제 테이블을 생성 및 유지하기 위한 비용으로 변환 대상 테이블의 총용량 수준의 추가확보가 필요하며, 이에 따른 비용을 연간 스토리지 자연증가율을 감안하여 TCO에 반영한다. 가장 중요한 것은 전산 프로세스를 운영하기 위한 CPU증설비용과 이에 따른 소프트웨어 사용비용 증가액을 산출하는 것이다.

따라서 본 논문에서는 유지 보수비용 중 CPU증설비용과 CPU용량에 따라 관련 소프트웨어 라이센스 비용을 책정하는 메인프레임 z/OS 운영체제 기반의 가격 정책에 근거해 CPU관련 비용 산출 기준을 제안해보고, 최소 CPU자원 사용을 통해 연계된 소프트웨어 사용비용을 제어하기 위한 실무적 방법을 제시해본다.

5.2.1 MIPS와 MSU

MIPS(Millions of Instructions Per Second)란, 일반적으로 CPU용량 및 CPU 처리능력의 측정기준을 의미하며 메인프레임 기반의 응용프로그램 실행과 연관된다. 메인프레임을 활용하는 대부분의 대형 기업들은 매년 15~20%씩 CPU자원 소비가 증가한다고 추정하고 있다[17].

메인프레임 사용량을 논의시 일반적으로 MIPS를 기준으로 추정하지만, 실제 측정값은 MSU(Million Service Units)로 측정한다. MSU란, z/OS 메인프레임 운영체제 기반에서 z/OS 1백만 서비스 단위에 대해 1시간동안 수행한 일의 양을 의미한다. MSU는 하드웨어 환경, 메모리 사용, 입출력 대역폭, 명령어 구성의 복잡성과 기타 여러 요소들을 포괄할 수 있도록 MIPS에 대한 대체 표시방식이며, 1MSU는 약 6MIPS이다[18].

L.M. Kwiatkowski 외(2010)는 MIPS관리 관점에서 MIPS 감소 프로젝트에 의해 영향을 받는 온라인 거래의 거래량에 대한 MSU의 거래당 평균 수준을 Fig.19.와 같이 약 37주간의 자료를 시계열로 분석하여 표시하였다.

즉, Fig.20.와 같이 도출된 MSU에 대해 6을 곱하여 MIPS로 전환한 후, 총 거래량의 MIPS를 산출하면 시스템의 여러 요소를 반영한 총자원사용량을 메인프레임 z/OS 시스템 용량의 지표인 MIPS로 산출할 수 있다.

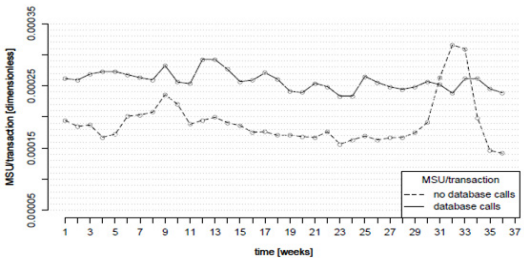

Fig. 20. Time series of the average weekly ratios of the MSUs to transactions volume for the transactions affected by the MIPS-reduction project[19].

5.2.2 CPU 단위 비용 산출 기준

Dr. Howard Rubin(2010)은 133개 회사와 21개 분야의 데이터를 분석하여 연간 1백만달러의 수익을 창출하는데 평균 0.37MIPS가 사용되며, 은행 및 금융분야는 0.98MIPS가 필요함을 보여주었다.

또한, 2010년 가트너의 'IT Key Metrics Data'를 사용하여 업종별 연간 1백만달러의 수익을 창출하는데 사용되는 MIPS 기준량 및 연간 1백억 달러의 매출을 지원하는데 필요한 MIPS 기준량을 도출하였고, 은행 및 금융 분야는 연간 매출 1백억 달러를 지원하기 위해 10,700MIPS가 필요하다고 하였다.

2010년 가트너의 'IT Key Metrics Data'는 1MIPS당 연간 운영비용을 평균 4,445달러로 산출하였다.

아래 Fig.21.은 가트너의 자료를 기준으로 해당 문헌에서 산출한 업종별 연간 1백만달러의 수익창출에 필요한 MIPS 기준량을 보여준다[20].

위의 결과를 토대로 은행 및 금융분야의 MIPS 사용량을 산출해보면, 연간 1백억달러의 매출을 창출하기 위해 필요한 CPU 자원은 10,700MIPS이다.

	Sector Averages n=133 companies				
	MIPS per $1M Revenue	Servers per $1M Revenue	Cost per MIPS (Source: Gartner 2010 Key Metrics)	Cost per Server (Source: 2010 Gartner Key Metrics)	Compute Cost (MIPS + Server Cost) per $1M Revenue
Banking	0.98	0.39	$ 4,445	$ 10,473	$ 8,441
Consumer Products	0.19	0.16	$ 4,445	$ 10,473	$ 2,520
Education	0.13	0.05	$ 4,445	$ 10,473	$ 1,102
Electronics	0.25	0.11	$ 4,445	$ 10,473	$ 2,263
Financial Services	1.07	0.46	$ 4,445	$ 10,473	$ 9,574
Food & Beverage Processing	0.18	0.12	$ 4,445	$ 10,473	$ 2,057
Government - Federal	0.49	0.12	$ 4,445	$ 10,473	$ 3,382
Government - State & Local	0.38	0.09	$ 4,445	$ 10,473	$ 2,632
Health Care	0.19	0.13	$ 4,445	$ 10,473	$ 2,206
Insurance	0.33	0.16	$ 4,445	$ 10,473	$ 3,143
Manufacturing	0.21	0.12	$ 4,445	$ 10,473	$ 2,190
Metals & Natural Resources	0.16	0.12	$ 4,445	$ 10,473	$ 1,968
Professional Services	0.14	0.08	$ 4,445	$ 10,473	$ 1,460
Telecommunications	0.85	0.25	$ 4,445	$ 10,473	$ 6,397
Transportation	0.23	0.21	$ 4,445	$ 10,473	$ 3,222
Utilities	0.16	0.08	$ 4,445	$ 10,473	$ 1,549
Cross Industry Average	0.37	0.17	$ 4,445	$ 10,473	$ 3,382

Fig. 21. The unit cost of MIPS and server for $1M revenue through the analysis of data from 21 sectors and 133 companies

Fig.21.의 1MIPS당 연간 운영비용 단가 4,445달러를 곱하면 해당 CPU자원비용은 약 4,756만달러로 산출된다.

또한, 은행 및 금융분야의 1백만달러의 수익 창출을 위해 연간 필요한 CPU 자원은 Fig.21.에서 보듯이 0.98MIPS 이므로, 이에 대한 CPU자원비용은 동일하게 1MIPS당 연간 운영비용 단가 4,445달러를 곱하여 약 4,356달러로 산출된다.

5.2.3 CPU 증설에 따른 유지 보수 비용 산출

데이터 보안통제 전산 프로세스 처리를 위한 CPU증설비용 산출을 위해 다음과 같이 연관 변수를 정의하고, 아래 공식에 따라 CPU증설비용 및 CPU증설에 따른 소프트웨어 사용비용 증가액을 산출한다.

- CPU usage qty: 제시한 전산프로세스 처리에 소요되는 CPU사용량(MIPS)
- Price per hour: CPU의 MIPS별 시간당 사용단가
- Avg runtime: 제시한 전산프로세스 1사이클(Cycle)에 대한 처리 소요시간
- Count per year : 제시한 전산프로세스의 1년간 작업횟수
- CPU usage ratio: 전체 CPU용량 대비 제시한 전산프로세스 처리를 위한 CPU사용비율
- CPU assigned ratio: 전체 CPU용량에 대해 제시한 전산프로세스 처리에 배분한 CPU사용배분비율
- Avg CPU available ratio per year: 전체 CPU용량 대비 1년 평균 유휴CPU가용율

> CPU증설비용 = CPU usage qty × Price per hour × Avg runtime × Count per year × (CPU usage ratio - CPU assigned ratio × Avg CPU available ratio per year)
> (단, 산출결과가 0보다 작으면 유휴가용자원의 범위 내에서 제시한 전산프로세스 처리가 가능한 수준으로 CPU증설비용은 0임,)

> CPU증설에 따른 소프트웨어 사용비용 증가액 = MIPS당 연간 소프트웨어 라이센스 비용 × CPU증설량(MIPS)

위의 공식에 근거하여 CPU증설비용을 0에 수렴하도록 하면, CPU증설용량에 비례하는 소프트웨어의 사용비용 증가액도 발생하지 않는다. 따라서 메인프레임 z/OS 운영체제 기반의 시스템을 사용하는 각 회사는 MIPS당 연간 운영비용을 산출해 위의 공식에 적용한다.

가트너의 2010년 발표자료는 MIPS당 연간 운영비용을 4,445달러로 계산하도록 제시하였다. 또 4.2.2의 Fig.18.에서 확인한 단위 전산 프로세스에 대한 총 수행시간 중 CPU time에 대해 각 회사의 CPU 운영용량에 따른 MSU로의 변환 보정치를 적용하여 MSU로 환산한 후, 1MSU당 6MIPS를 적용한 MIPS를 산출하면 데이터 보안통제 전산 프로세스의 총 CPU사용량을 확인할 수 있다.

이렇게 산출된 전산 프로세스의 총CPU사용량이 회사의 유휴가용자원에 대한 CPU사용배분율의 범위 내에서 처리가 가능한 경우 CPU증설비용은 추가발생하지 않는다. 본 논문에서 테스트를 진행하였던 A금융사의 유휴CPU자원을 확인하기 위해 조사한 자료는 아래와 같다.

- 테스트시스템의 상황별 CPU사용율
  - 월말 영업시간: 90~100%
  - 기타 영업시간: 60~70%
  - 비영업시간: 30~35%
- 테스트시스템의 한계CPU사용율: 100%
- 유휴CPU가용율 = 한계CPU사용율 - 제시한 전산프로세스가 수행되는 시간대의 전체 CPU사용율

A금융사는 CPU사용율이 100%가 도달하기 전에 별도 전산작업을 통제하지 않고, 수행 중인 전산작업의 CPU사용수준만 통제하므로, 한계CPU사용율을 100%로 설정하였으며, A금융사가 비영업시간에 데이터 보안통제 전산 프로세스를 처리하도록 배정하는 경우 유휴CPU가용율은 약 65~70%수준으로 이 범위 내에서 배정한 전산 프로세스의 처리가 완료된다면, CPU증설비용 및 CPU증설에 따른 소프트웨어 사용 증가 비용은 발생하지 않는다.

일반적으로 데이터 보안통제 전산 프로세스는 변환 대상 전체 테이블을 대상으로 하나, 변환 미확인 데이터의 검증이 필요한 최종 테이블의 용량이 크지 않은 경우에는 CPU자원의 과도한 사용이 발생하지 않는다.

그러나 해당 테이블이 대용량인 경우 물리적인 분할이 적용된 경우가 많으며, 이 경우 변환 미확인 데이터 추출작업 시 CPU자원을 과도하게 사용하게 될 수 있으므로, 물리적인 분할 기준의 작업수행 배정, 동시 작업 처리개수 등에 대한 최적화된 워크로드 밸런싱을 통해 유휴CPU가용 범위 내에서 제한된 시간 내에 처리가 가능하도록 하여 CPU증설비용이 발생하지 않도록 하는 것이 매우 중요하다.

5.3 비용편익 산출

데이터 보안통제 프로세스의 효과성을 분석하기 위한 비용편익(Cost Benefit Analysis: CBA)을 다음과 같이 산출한다[21].

> 데이터 보안통제 프로세스의 비용편익 = (변환 미확인 데이터의 노출에 따른 자산 가치의 피해액 / 데이터 보안통제 프로세스의 TCO) × 100

5.1에서 A카드사의 추정된 자산 가치 피해액에 대한 데이터 보안통제 프로세스의 TCO를 아래와 같이 가정하여 비용편익을 산출하면 다음과 같다.

- 자산 가치의 피해액: 1,522억원
  - 영업손실금액 445억원
  - 후속처리비용 217억원
  - 정신적 손해배상비용 860억원
- 초기 개발비용: 3억원
- 스토리지 비용(5년 TCO 가정): 약 24.48억
  - 변환 대상 테이블 용량: 7TB 가정
  - 하이엔드급 스토리지: TB당 1억원 적용
  - 스토리지 구성: RAID1[12] 방식 적용
  - 연간 스토리지 자연증가율: 15% 적용

위 사례는 가용 시간 내 데이터 보안통제 전산 프로세스를 수행하기 위해 최적의 성능을 확보한 하이엔드(High-End)급 스토리지를 도입하고, 장애 시 복제테이블의 신속한 복구처리 및 전산 프로세스 수행 시의 I/O 효율화를 위해 RAID1 레벨의 스토리지 구성방식 채택 및 TB당 도입비용을 약 1억원 수준으로 가정하였다. RAID1 구성방식은 저장대상 용량의 2배 수준의 물리적 용량이 필요하므로, 위 사례에서 변환 대상 테이블 용량이 7TB인 경우 필요한 물리적 스토리지 용량은 14TB이다.

따라서 14TB에 대해 최초도입시점 익년도 부터 4년간 연간 자연증가율 15%를 적용하면, 스토리지의 5년 TCO는 약 24.48억원이 산출된다.

만약, A카드사가 전산 프로세스의 수행시간이 충분히 확보되어, 스토리지 구성방식을 데이터비중 75%, 패리티(parity)비중 25%의 RAID5[13] 레벨로 구성한다면, 필요한 스토리지 용량은 약 9.33TB가 되므로, 이에 대한 5년 TCO는 약 16.32억원이 산출된다. 또한, IDC의 조사결과처럼 하이엔드급 스토리지의 수요는 지속적으로 낮아지고 있어, 단위당 도입비용은 낮아지고 있는 추세이다[22].

이에 따라, 현 추세대로라면 향후 동일 용량에 대한 하이엔드급 스토리지 비용은 현재보다 낮아질 것으로 예상된다. 따라서 A카드사가 유휴가용자원에 최적화된 전산작업의 워크로드 밸런싱을 통해 CPU 증설비용과 소프트웨어 사용비용 증가액을 발생시키지 않았다고 가정하면, 스토리지 비용을 보수적으로 적용하여 위의 자료들로 산출한 A카드사의 데이터 보안통제 프로세스의 총 TCO는 약 27.48억원이며, 비용편익은 약 5,538%가 산출된다.

물론 이렇게 과도하게 높게 산출된 비용편익은 2014년 유례없는 개인정보유출 사고를 근거로 해 산출된 것이지만, 관리적 보안의 미비로 카드3사의 개인정보유출 사건과 같은 문제가 또다시 발생하기 전에 데이터 보안통제 프로세스를 적용해야 하는 중요성이 매우 크다는 것을 의미하며, 단위당 스토리지 가격이 향후 현재보다 낮아진다면 데이터 보안통제 프로세스의 총 TCO는 지속적으로 낮아질 것으로 예상되어 실효성은 더욱 높아질 것이다.

12) 여러 개의 하드디스크에 데이터를 중복하여 분리 저장하는 기술인 RAID(스토리지 구성 방식 중에, 패리티가 없는 미러링 방식의 RAID 구성 레벨이다. 물리적 디스크의 실사용율은 약 50%이다.

13) 패리티를 각 디스크마다 분산하여 데이터부분과 저장하는 RAID구성방식이며, 몇 개의 디스크로 RAID5 레벨을 구성할지 결정해야 한다. 물리적 디스크의 실사용율은 디스크 구성개수를 n이라 하면, (n-1)/n 의 비율이다.

VI. 결론 및 향후 연구방향

금융회사는 복잡한 비즈니스 구조 속에서 양질의 테스트 결과를 확보하기 위해 운영시스템과 유사한 수준의 테스트시스템을 구성해야 하나, 감독기관의 컴플라이언스 준수를 위해 테스트시스템에서 개인식별정보는 변환하여 운영한다.

본 논문에서는 이러한 기업의 컴플라이언스의 준수 노력에도 불구하고 컴플라이언스 위반의 취약성을 높이는 의도치 않게 테스트시스템에 유입된 변환 미확인 데이터의 노출수준을 낮추는 대책을 제시하였다.

또한 제시한 대책을 전산 프로세스로 구현하고, 각각의 알고리즘에 대한 테스트를 진행하여 본안을 적용하려는 기업에 워크로드 밸런싱을 통한 의사결정에 도움이 되는 테스트 케이스를 제공함으로써 실효성 있는 검증을 진행하였다.

그리고 이러한 데이터 보안통제 프로세스를 도입 시 해당 원인에 따른 개인정보유출 차단 관점에서 효과성을 분석해보고, 최소한의 TCO로 최적화된 자원을 사용하여 본 논문에서 제시한 전산 프로세스의 적용을 검토할 수 있도록 하였다.

본 논문에서 살펴본 테스트시스템의 개인식별정보 변환 미확인 데이터의 노출은 감독기관 규정 위반의 취약수준 증가, 관리적 보안 미비 등에 따른 정보유출 취약수준 증가 등의 문제의 원인이 되므로 그 사안이 중대하나, 현재 문제해결을 위한 많은 논의가 이루어지지 않았다.

따라서 본 논문에서 제안한 데이터 보안통제 프로세스를 각 기업의 자원상황에 맞게 최적화하여 적용하고, 메인프레임 기반 외의 다른 시스템에도 해당 환경에 맞게 커스터마이징하여 확대 적용해 나간다면, 업무 프로그램 개발 및 운영에 따라 의도치 않게 발생하는 테스트시스템의 개인식별정보 변환 미확인 데이터 보유 가능성을 없앰으로써 예상치 못한 정보유출사고의 위험 및 감독기관의 규정 위반을 예방하는 데 큰 도움이 될 것이다.

References

〔1〕 The total estimated damage due to leakage of the credit card company is 100 billion won, http://view.asiae.co.kr/news/view.htm?idxno=2014012711034390924

〔2〕 Regulations for Electronic Banking Supervision of the Financial Supervisory Service, http://www.law.go.kr/행정규칙/전자금융감독규정

〔3〕 Regulations of the Privacy Act, http://www.law.go.kr/DRF/lawService.do?OC=illusfac&target=law&MST=136728&type=HTML

〔4〕 Definition of the system 'z/OS', http://en.wikipedia.org/?title=Z/OS

〔5〕 Definition of the DBMS 'DB2', http://en.wikipedia.org/wiki/IBM_DB2

〔6〕 What is a DBMS log?, http://www.answers.com/Q/What_is_a_DBMS_log

〔7〕 What is the System Catalog?, http://www.informit.com/library/content.aspx?b=STY_Sql_24hours&seqNum=170

〔8〕 Definition of the Referential Integrity, http://ko.wikipedia.org/wiki/%EC%B0%B8%EC%A1%B0%EB%AC%B4%EA%B2%B0%EC%84%B1

〔9〕 Definition of the Referential Constraints, http://www.sapdb.org/7.4/htmhelp/6d/117c5fd14811d2a97400a0c9449261/content.htm

〔10〕 What are Entity Relationships Diagrams?, http://www.smartdraw.com/resources/tutorials/entity-relationship-diagrams/

〔11〕 What is 'CRUD'?, http://ko.wikipedia.org/wiki/CRUD

〔12〕 What is 'primary key'?, http://terms.naver.com/entry.nhn?docId=851114&cid=42346&categoryId=42346

〔13〕 What is 'Load Balancing'?, http://en.wikipedia.org/wiki/Load_balancing_(computing)

〔14〕 Estimated loss of three of credit card companies which flow out PII, http://www.hankyung.com/news/app/newsview.php?aid=201405192887g

〔15〕 Jong-hwan Kim, Jong-in Lim, "Composition and Policy Direction of Compensation Insurance Against Customer Information Infringements in Financial Transactions," The Journal of Society for

e-Business Studies, pp.7-8, Aug. 2014
〔16〕 Definition of TCO, http://terms.naver.com/entry.nhn?docId=839269&cid=42344&categoryId=42344
〔17〕 L.M. Kwiatkowski & C. Verhoef, "Reducing operational costs through MIPS management," Department of Computer Science, Vrije Universiteit Amsterdam, pp.1-2, Mar. 2010
〔18〕 L.M. Kwiatkowski & C. Verhoef, "Reducing operational costs through MIPS management," Department of Computer Science, Vrije Universiteit Amsterdam, pp.6-7, Mar. 2010
〔19〕 L.M. Kwiatkowski & C. Verhoef, "Reducing operational costs through MIPS management," Department of Computer Science, Vrije Universiteit Amsterdam, pp.8, Mar. 2010
〔20〕 Dr. Howard Rubin, "Economics of Computing -The Internal Combustion Mainframe 〔Expanded Version〕," Technology Economics, pp.1-2, 2010
〔21〕 What is 'Cost Benefit Analysis'?, http://en.wikipedia.org/wiki/Cost%E2%80%93benefit_analysis
〔22〕 Weak High-End demand results in world-wide external disk storage systems revenue falling at rates not seen since 2009, according to IDC, http://www.idc.com/getdoc.jsp?containerId=prUS24914414

〈저자소개〉

최 영 진 (Yeong jin Choi) 정회원
1997년 2월: 서경대학교 컴퓨터공학과 학사
2013년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 금융정보보안, 위험관리, 정보보호 및 개인정보보호정책

김 정 환 (Jeong-hwan Kim) 정회원
2007년 2월: 국민대학교 경영학부(재무.금융전공) 학사
2013년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 금융정보보안, 위험관리, 데이터베이스, 어플리케이션 성능튜닝

이 경 호 (Kyung Ho Lee) 종신회원
1989년 8월: 서강대학교 수학과 학사
1997년 8월: 서강대학교 정보통신대학원 석사
2009년 8월: 고려대학교 정보보호대학원 박사
1994년 2월~현재: 삼성그룹, 네이버(주), 시큐베이스 등 근무
2011년 9월~현재: 고려대학교 정보보호대학원 조교수
〈관심분야〉 위험관리 정보보호컨설팅 정보보호 및 개인정보보호정책