韓國 컴퓨터情報學會 論文誌
第10卷 第5號, 2005. 11.

2005-10-5-1-8

트랜잭션 순서 기반 보안 동시성 제어 기법

이 원 섭*, 이 상 희**

Transactions Ordering based Secure Concurrency Control Scheme

Won-Sup Lee*, Sang-Hee Lee**

요 약

각기 다른 보안 등급을 가지고 트랜잭션을 실행하는 다중 등급 보안 데이터베이스관리시스템에서의 보안 동시성 제어 기법은 비밀통로에 대한 고려가 있어야 한다. 보안 동시성 제어에 관한 선행 연구들은 동시성 제어 과정에서의 기밀성은 보장하고 있지만 무결성과 가용성에 대한 고려를 간과하고 있다. 본 논문에서는 검증된 트랜잭션 순서 관계를 사용하여 각기 다른 보안 등급의 트랜잭션들 간에 형평성을 제공하는 보안 동시성 제어 기법을 제안한다.

Abstract

While the secure concurrency control schemes in multilevel secure database management systems synchronize transactions cleared at different security level they must consider the problem covert channel. although previous works achieve the confidentiality successfully, they overlook the integrity or the availability. For being evaluated as highly secure database systems , the multilevel secure database management systems must achieve the confidentiality, integrity, and the availability that are the well-known major security aspects. By use of verified transactions ordering relationship, in this paper, we propose a new secure concurrency control scheme that is capable of increasing the degree of fairness among transactions cleared at different security levels.

▸ Keyword : 데이터베이스(database), 보안(security), 동시성제어(concurrency control)

• 제1저자 : 이원섭 • 교신저자 : 이상희
• 접수일 : 2005.10.10, 심사완료일 : 2005.11.09
* 인덕대학 컴퓨터전자전공 조교수, ** 청강문화산업대학 컴퓨터소프트웨어과 교수

Ⅰ. 서 론

데이터에 대한 효율적 이용과 일관성을 유지하기 위해 데이터를 중앙집중식으로 관리하는 데이터베이스 시스템에서 데이터에 대한 트랜잭션의 동시 처리는 일반적이다. 이런 다중 트랜잭션 실행 환경에서 데이터베이스가 일관성을 유지하기 위해서는 공유 데이터에 대한 읽기 혹은 쓰기 연산 처리가 동시성 제어 기법이라는 통제된 방법에 의해서 실행되어야만 한다[1, 2]. 기존의 보안을 고려하지 않은 동시성 제어 기법은 공유 데이터에 접근하는 동시 실행되는 트랜잭션들 중에서 특정 트랜잭션을 지연시키거나 철회하는 방법으로 데이터베이스의 일관성을 유지한다.

하지만, 특정 데이터에 대한 인가되지 않은 사용자들의 임의 접근을 막기 위해서는, 트랜잭션 및 데이터에 대해 다중 등급 비밀 등급을 부여하고 그 비밀 등급에 허용된 처리만을 수행하도록 하는 다중 등급 보안 데이터베이스 시스템(multilevel secure database management system: MLS/DBMS)의 경우, 동시성 제어과정에서 발생하는 트랜잭션 실행 지연이나 철회에 의해 발생하는 현상을 이용한 비밀경로(covert channel)를 막을 수 있는 보안 동시성 제어 기법(secure concurrency control)이 필요하다[3, 4].

동시성 제어 과정에서 발생하는 비밀경로를 제거하기 위하여 많은 선행연구들이 [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] 제안되었다. 이러한 선행연구들은 보안을 유지하기 위해서는 데이터베이스 시스템(database system:DBS)의 무결성 혹은 가용성을 위반하는 것이 불가피하다는 관점을 가지고 있다. 선행연구들을 무결성 혹은 가용성 위반의 관점에서 분류할 경우 아래와 같다.

[정확성 평가 방법]
- 트랜잭션 완료/철회 판단 사용자 위임 방법[7, 11, 19]
- 비제한적 버전 관리 방법[17, 19]
- 정확성 저하 허용 방법[10, 16, 17, 20, 21]
- 다단계 보안등급 트랜잭션 방법[5, 6, 9, 12, 13]
  가용성 침해 허용 방법
- 하위 보안등급 트랜잭션 우선 방법[7, 8, 11, 15, 19]
- 제한적 버전관리 방법[11, 14]
- 연 속철회 허용 방법[18]

정확성 평가 방법에 따른 동시성 제어 결과는 비제한적 버전 관리 방법[17, 19]만이 DBS의 정확성을 보장한다. 가용성 침해 허용 방법은 연산들 간에 충돌이 발생할 경우 상위 비밀등급 트랜잭션을 일방적으로 지연이나 철회함으로 인한 상위 비밀등급 트랜잭션의 가용성 침해를 야기한다. 보안성의 구성요소를 기밀성, 무결성, 가용성으로 볼 때, 무결성이나 가용성에 대한 침해가 특정 트랜잭션집단의 의도에 의해 야기된다면 전체적인 보안관점에서는 보안성 침해로 해석될 수 있다[23, 24, 25, 26, 27].

본 연구에서 제시한 트랜잭션 순서 기반의 보안 동시성 제어(transaction ordering based secure concurrency control: TOSCC) 기법은 다중 버전 데이터베이스를 기반으로 트랜잭션들이 제기한 읽기연산은 거부되지 않는다. 따라서 단일버전 데이터베이스 기반의 동시성 제어기법들보다 트랜잭션 철회나 대기를 현저히 줄일 수 있다. 또한 비밀경로 차단을 통하여 기밀성과 엄격한 직렬화 가능성 유지를 통해서 무결성을 유지하며, 상이한 보안등급의 트랜잭션 그룹의 가용성 저하를 방지한다. 더불어 불필요하다고 판단되어진 데이터 버전들을 제거함으로써 데이터베이스의 크기가 과도하게 증가하는 것을 방지할 수 있다.

Ⅱ. 보안모형

TOSCC는 보안성과 직렬화 가능성을 유지하기 위하여 새로운 보안모델을 채택하였다. 제시된 보안모델은 일반적인 정보보안 시스템에 적용할 수 있는 보편성을 가지고 있는 모델이다.

보안 데이터베이스 시스템의 구성요소는 크게 주체와 객체로 나눌 수 있는데 주체는 정보처리를 위하여 능동적으로 정보를 보유하고 있는 대상을 접근하는 개체를 의미하고 객체는 주체에 의해서 접근되어지는 정보를 보유하고 있는 개체를 의미한다. 즉, 주체는 사용자, 트랜잭션 등이 될 수 있으며, 객체는 데이터 항목, 데이터 파일 등이 그 대상이 될 수 있다. 보안 데이터베이스 시스템에서의 주체들 중 신뢰성을 보장할 수 있는 주체는 보안관리자에 한정한다.

[가정 1]: 신뢰 가능한 보안 동시성 제어기
- 보안 동시성 제어기의 동작은 자체적으로 보안성을 침해할 개연성이 없는 것으로 가정한다.

[가정 2]: 트랜잭션의 비신뢰성
- 보안관리자를 제외한 사용자들이 발생시킨 트랜잭션은 보안 침해의 개연성을 항시 보유한 것으로 가정한다.

[가정 3]: 갱신연산의 신뢰성
- 트랜잭션은 신뢰할 수 없지만 트랜잭션들이 제기하는 갱신연산은 보안성을 침해하지 않는 것으로 가정한다.

트랜잭션이 어느 특정 데이터를 접근하고자 할 때 강제적 접근제어를 적용한다. 강제적 접근제어가 적용됨에 따라 모든 트랜잭션들의 데이터에 대한 접근은 트랜잭션과 데이터에 부여된 보안 등급에 의거하여 이루어진다. 이러한 접근제어는 Bell-LaPadula 모형을 기반으로 하여 아래와 같이 정의된 특성으로 이루어진다.

[정의 1]: 상위 보안등급 판독금지
- 트랜잭션 Ti의 보안 등급을 CL(Ti), 데이터 x의 보안등급을 CL(x)라고 할 경우 CL(x) > CL(Ti)일 경우 Ti의 x에 대한 판독연산은 허용되지 않는다.

[정의 2]: 상향/하향 보안등급 갱신금지
- 트랜잭션, (Ti)의 보안 등급을 CL(Ti), 데이터 x의 보안등급을 CL(x)라고 할 경우 CL(x) = CL(Ti)인 경우에 한하여 Ti의 x에 대한 갱신을 허용한다.

Bell-LaPadula에 의거한 보안정책이 운영되어지기 위해서는 반드시 보안관리자에 의하여 일관성 있게 각 데이터 항목과 사용자에 대하여 보안등급이 부여되어져야 한다. 이 규칙이 적용되면 데이터베이스 시스템은 강제적 접근제어(Mandatory Access Control:MAC)를 적용한 보안정책 운영이 가능하게 된다.

하지만, 강제적 접근제어가 적용되어지더라도 지연이나 철회와 같은 간섭현상으로 인한 비밀경로의 완전한 제거는 불가능하다. 하지만, 트랜잭션들간의 간섭현상을 신중히 처리할 수 있는 동시성 제어기법을 적용할 경우에는 비밀경로를 충분히 제거할 수 있다.

III. 제안 보안 동시성제어 기법

본 연구는 직렬화 가능성을 위반할 가능성이 있는 충돌들도 반드시 직렬화 가능성을 침해하지는 않는다는 사실에 주목하고 있다. 기존의 비관적 동시성 제어기법에서는 직렬화 가능성을 침해할 가능성에 너무 많은 비중을 부여함으로써 불필한 간섭을 발생시켰다. 기존의 낙관적 동시성 제어기법에서는 결국에는 철회되어질 연산들을 불필요하게 완료 직전 시점까지 수행하는 문제점이 있다.

이런 문제점을 해결하기 위해 TOSCC는 트랜잭션의 각 연산을 처리하는 과정에서 해당 연산이 직렬화 가능성을 침해하고 있는지를 검증하여 실제 침해하는 것으로 확인되는 경우에만 철회함으로써 불필요한 간섭과 불필요한 연산의 수행을 제거한다. 하지만, 비밀경로를 제거하기 위해서는 비밀경로를 구성할 수 있는 개연성이 있는 판독연산에 대하여 대체 데이터 버전을 판독하도록 한다. 먼저, 비밀경로를 제거하기 위하여 규칙 1을 제시한다.

[규칙 1]: 비밀경로 제거
- CL(Ti) > CL(Tj) 인 경우 Ti에 의한 Tj의 간섭은 반드시 제거되어야 한다.

TOSCC는 트랜잭션들 간의 충돌을 동일 보안 등급 부여시로 한정하는 방법을 채택하여 상위 보안등급 트랜잭션군에 편중된 철회나 지연을 방지하고 있다.

각 연산 단위에서의 능동적인 검증을 가능하게 하기 위하여 TOSCC는 트랜잭션들 간의 순서관계를 아래의 규칙 2에 의거하여 관리한다.

[규칙 2]: 트랜잭션 실행 순서
(가) r[x]가 등장할 경우 Tj는 Ti에 선행한다.
(나) x의 버전, xi와 xj에 대하여 xi가 xj보다 먼저 데이터베이스에 기록되었을 경우 Ti는 Tj에 선행하며 xi와 xj의 순서관계는 xi ≪ xj로 표시한다.
(다) xi ≪ xj인 상태에서 r[x] ≪ w[x]가 등장할 경우 Ti는 Tk에 선행한다.

규칙2를 엄격히 유지, 관리하는 목적은 임의의 두 트랜잭션들 간에 최초로 설정된 순서관계를 트랜잭션이 완료할 때까지 변경됨이 없이 유지하기 위해서이다. 최초에 설정된 트랜잭션 순서관계가 변경된다는 것은 최초에 선행하는 것으로 설정된 트랜잭션이 어느 시점에 선행과 동시에 후행하게 된다는 것을 의미하게 되며 이는 직렬화가능성을 위반하는 직접적인 원인을 제공하게 된다.

직렬화 가능성을 유지하기 위해 TOSCC는 Ti와 Tj에 대하여 선행 트랜잭션 집합(Preceding Set Transactions : PST)과 후행 트랜잭션 집합(Following Set of Transactions: FST)을 유지한다. Ti가 Tj를 선행할 경우 트랜잭션 순서관계는 Ti → Tj로 표시하며, Ti→Tj인 경우 PST(Tj)는 Ti를 원소로 가지며, FST(Ti)는 Tj를 원소로 가진다. TOSCC는 선행/후행트랜잭션 집합을 운영함에 있어 규칙3을 엄격히 준수함으로써 직렬화가능성을 유지한다.

[규칙 3]: 선행/후행 트랜잭션집합의 분리

- 임의의 트랜잭션 Ti에 대하여 Ti가 제기한 모든 연산들은 PST(Ti) ∩ FST(Ti)가 항상 공집합이 되도록 한다.

하지만 규칙 3을 실현하기 위하여 TOSCC는 대체판독 기능을 가져야 한다. 대체판독 기능을 가져야 되는 불가피성에 대하여 예 1에서 설명한다.

[예 1]: 대체 판독의 필요성

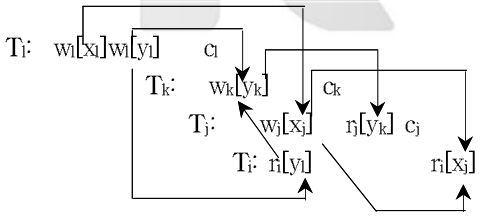

그림 1. 최종 데이터 버전 판독에 따른 직렬화가능성 위반
Fig1. Serializability violation
by reading latest data version

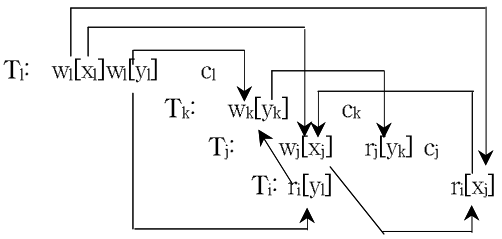

그림 2. 대체 데이터 버전 판독에 따른 직렬화가능성 유지
Fig2. Serializability preservation by reading appropriate data version

(그림 1)에서 나타나듯이 ri[x]를 제기한 트랜잭션 Ti가 xj를 판독할 경우 트랜잭션순서관계는 Ti → Tk → Tj → Ti가 된다. 이에 따라 PTS(Ti) ∩ FST(Ti)는 {Tj, Tk}로 구성되며 이는 규칙 3을 위반하여 Ti, Tj, Tk간의 직렬화가능성을 더 이상 유지하지 못하게 된다.

(그림 1)의 문제는 (그림 2)에서 보여지듯이 Tj로 하여금 트랜잭션 Ti이 생성한 xi을 판독할 경우 즉, ri[xj] 대신 ri[xi]을 수행하도록 함으로써 Ti와 Tj간의 순서관계를 변경됨이 없이 유지하여 규칙3을 준수할 수 있다. 그 결과 직렬화가능성도 유지할 수 있게 된다.

예 1과 같이 주어진 판독연산을 제기한 트랜잭션에게 특정 데이터 버전을 제공하는 것이 직렬화가능성을 위반하는 것으로 확인될 경우 선정된 데이터 버전을 선행하고 있는 데이터 버전을 대체 제공한다.

직렬화가능성 위반사실을 판독연산 도착 시에 발견된다면 대체 데이터 버전 판독으로 직력화가능성을 유지할 수 있겠지만 갱신연산 도착 시에 발견된다면 갱신연산을 철회시켜야만 직렬화가능성을 유지할 수 있을 것이다. 이에 관한 경우를 예 2에서 제시한다.

[예 2]: 비밀경로 생성 개연성 제거

CL(Ti) > CL(Tk)인 것으로 가정할 경우, (그림 3)에서 wk[z]가 처리된다면 반드시 zk가 새로운 데이터 버전으로 생성되어야 한다. 이 경우 트랜잭션순서관계는 Ti → Tk → Tj → Ti가 될 것이다. 따라서 PTS(Ti) ∩ FST(Ti)는 {Tj, Tk}로 구성되며 이는 규칙 3을 위반하여 Ti, Tj, Tk 간의 직렬화 가능성을 유지될 수 없다. 직렬화가능성을 유지하기 위하여 wk[zk]를 거부하여 Tk를 철회할 경우 규칙 1를 위반하게 되어 Ti와 Tk사이에 비밀경로를 생성하게 된다. 결국 wk[zk]를 수행하여야 하는 시점에서는 직렬화가능성과 비밀경로 제거를 동시에 이룰 수는 없게 된다.

(그림 4)에서와 같이 비밀경로 생성 가능성이 있는 판독 연산 ri[x]로 하여금 xj대신 대체 데이터 버전 xl을 판독토록 함으로써 순서관계를 Tl → Ti → Tk →Tj, Tl → Tk → Tj 그리고 Tl → Tj 로 구성될 수 있도록 하였다. 이경우 wk[zk]의 수행이 Ti → Tk의 순서관계를 변경시키지 않게 되며 그 결과 직렬화 가능성과 비밀경로를 동시에 제거할 수 있게 된다.

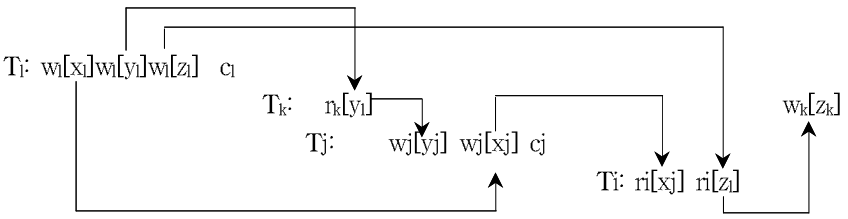

그림 3. 갱신연산의 처리에 의한 직렬화가능성 위반
Fig3. Serializability violation by processing update operations

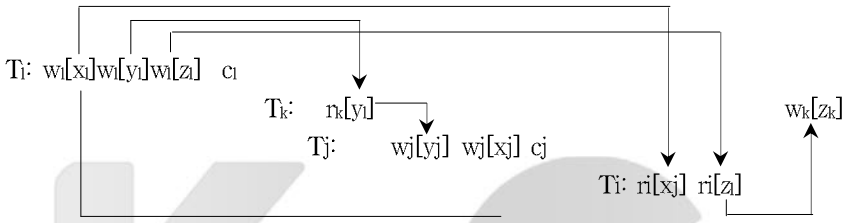

그림 4. 대체 데이터 버전 판독에 의한 비밀경로 개연성 제거
Fig 4. Elimination of probable covert channel by reading appropriate data version

규칙 4는 직렬화 가능성 유지와 비밀경로 제거를 동시에 달성할 수 있는 대체 버전 판독의 필요성을 규칙으로 제시한 것이다.

[규칙 4]: 보안적 데이터 버전 선정

(가) 트랜잭션 Ti가 제기한 판독연산에 대한 적절한 데이터 버전이 되기 위해서는 데이터 버전의 선정이 Ti에 대한 선행/후행트랜잭션집합의 분리 규칙의 분리 규칙을 준수할 수 있어야 한다.

(나) 트랜잭션 Ti가 제기한 판독연산 ri[x]에 대하여 규칙 2의 (가)에 의해 xj가 적절한 데이터 버전으로 선정되었다고 가정한다. 이때, PTS(Tj)에 CL(Ti) > CL(Tk)인 현재 수행중인 트랜잭션 Tk가 있을 경우 xj는 ri[x]에 대한 적절한 데이터 버전이 되지 못한다. 이는 향후 발생할 수도 있는 Tk의 갱신연산에 의하여 비밀경로가 발생할 수 있거나 혹은, 비밀경로를 방지하는 과정에서 직렬화가능성을 위반할 수 있는 가능성이 있기 때문이다. 따라서 PTS(Ti)이 Tk를 포함하고 있지 않은 데이터 버전 xl을 ri[x]를 위한 대체 데이터 버전으로 선정한다.

TOSCC는 다른 트랜잭션들과 충돌을 발생시키는 연산의 수행요청을 받을 때마다 연산을 제기한 트랜잭션을 기준으로 선행 및 후행트랜잭션집합을 분리 여부를 조사하여 직렬화 가능성을 검증한다. 이 과정에서 트랜잭션마다의 선행 및 후행트랜잭션집합을 탐색하고 검증하는데 처리비용이 발생한다. 따라서 선행 및 후행트랜잭션집합에 대한 처리 부담을 감소시키기 위해서는 선행 및 후행트랜잭션집합을 최소로 유지하는 것이 필요하다. 이러한 경량화는 현재 각 트랜잭션들의 선행 및 후행트랜잭션집합에 등록되어진 트랜잭션들 중에서 더 이상의 유지 및 관리가 무의미한 트랜잭션들을 찾아내어 제거하기 위하여 TOSCC는 규칙 5를 적용한다.

[규칙 5]: 무가치 트랜잭션 삭제

(가) 트랜잭션, Ti,가 완료하는 시점에서 Ti가 유일하게 수행중

인 트랜잭션인 경우 FST(Ti)에 등장하는 모든 트랜잭션들은 무가치 트랜잭션으로 판단되어 트랜잭션순서관계에서 삭제된다.

(나) 트랜잭션 Ti가 완료하는 시점에서 FST(Ti)에 존재하는 임의의 이미 완료된 트랜잭션, Tk의 SPT(Tk)에 현재 수행중인 트랜잭션이 전혀 포함되어 있지 않을 경우 Tk는 무가치 트랜잭션으로 판단되어 트랜잭션순서관계에서 삭제된다.

(다) 트랜잭션 Ti가 완료하는 시점에서 PST(Ti)에 존재하는 임의의 완료된 트랜잭션, Tk,의 FST(Tk)에 현재 수행중인 트랜잭션이 전혀 포함되어 있지 않을 경우 Tk는 무가치 트랜잭션으로 판단되어 트랜잭션 순서 관계에서 삭제된다.

다중 버전 데이터베이스 시스템의 경우 필연적으로 발생하는 문제는 한 데이터 항목에 대하여 다중 버전의 객체들을 가짐에 따른 데이터베이스 용량이 증가에 있다. 이는 기억장치를 접근하는데 물리적 지연을 가져오게 되며 이는 정보시스템 전체에서 병목현상으로 나타날 수 있다. 따라서 더 이상의 보존이 무의미하다고 판단되어지는 데이터 버전들은 능동적으로 삭제하는 것이 바람직하지만, 무가치 데이터 버전 삭제작업이 경우에 따라서는 특정 트랜잭션이 반드시 판독하여야 하는 데이터 버전이 존재하지 못하게 되는 경우를 발생시킬 수 있다.

[규칙 6]: 무가치 데이터 버전 삭제

(가) 현재 신규로 생성되고 있는 데이터 버전은 무가치 데이터 버전 삭제의 대상에서 제외한다.

(나) 두 개의 인접한 데이터 버전 xa와 xb가 xa ≪ xb,의 데이터 버전 순서관계를 가지고 두 개의 수행중인 트랜잭션들의 집합 STi와 STj가 있다고 가정할 때, STi와 STj에 있는 모든 트랜잭션들은 xa와 xb의 생성 트랜잭션인 Ta와 Tb에 대하여 트랜잭션 순서 관계를 가진다고 가정한다. 이러한 가정 하에서 아래의 조건을 만족할 경우 xa는 무가치 데이터 버전으로 선정되어 삭제의 대상이 된다.

(나-1) xa와 xb가 STi와 STj에 포함되어 있는 모든 트랜잭션들에 대하여 판독 가능한 데이터 버전인 경우

(나-2) xa와 xb가 STi와 STj에 포함되어 있는 모든 트랜잭션들에 대하여 판독 불가능한 데이터 버전인 경우

(나-3) xa와 xb가 STi에 포함되어 있는 모든 트랜잭션들에 대하여 판독가능하고 xa와 xb가 STj에 포함되어 있는 모든 트랜잭션들에 대하여 판독 불가능한 경우

(나-4) xa와 xb가 STi에 포함되어 있는 모든 트랜잭션들에 대하여 판독 불가능하고 xa와 xb가 STj에 포함되어 있는 모든 트랜잭션들에 대하여 판독 가능한 경우

무가치 데이터 버전 삭제 규칙을 적용할 경우 TOSCC를 채택한 MLS/DBMS는 데이터 항목당 데이터 버전을 일정수로 유지할 수 있다. 따라서 시간이 경과함에 따라 데이터베이스의 용량이 증대함에 따른 기억장치 접근시간이 선형적으로 증대하는 기존의 다중 버전 데이터베이스 기반 보안 동시성 제어 기법보다 향상된 성능을 기대할 수 있다.

IV. 정확성 증명

[정리 1]: TOSCC는 직렬화가능성을 보장한다.

[증 명]: 규칙2를 엄격히 유지, 관리하여 임의의 두 트랜잭션들 간에 최초로 설정된 순서관계를 트랜잭션이 완료할 때까지 변경됨이 없이 유지하고, 규칙 3을 만족할 경우 트랜잭션 Ti를 선행하는 트랜잭션은 결코 Ti를 후행하지 않게 되며 이는 트랜잭션 Ti를 기준으로 할 경우 직렬화가능성을 보장할 수 있음을 의미한다. 따라서 TOSCC에 의하여 제어되는 모든 트랜잭션들의 연산들에 대하여 규칙 3을 적용할 경우 TOSCC는 직렬화가능성을 보장할 수 있게 된다.

[정리 2]: TOSCC에 의한 거래이력은 보안성을 보장한다.

[증 명]: TOSCC가 다중 버전 데이터베이스를 기반으로 하고 있으므로 트랜잭션의 철회는 반드시 갱신연산에 의해서만 야기된다. 또한 규칙 3에 의거하여 동시성 제어를 수행할 경우 선행/후행트랜잭션집합의 분리 규칙을 위반하는 트랜잭션들은 반드시 동일한 보안등급을 부여받은 경우로 한정됨에 따라 상이한 보안 등급 트랜잭션들 간의 간섭현상을 근원적으로 제거하게 된다. 따라서, TOSCC의 트랜잭션조정 결과는 직렬화가능성의 획득에 따른 데이터베이스의 무결성을 유지하며 간섭현상을 동일한 보안등급 트랜잭션들 간으로 한정함에 따라 비밀경로를 제거할 있게 된다. 그 결과 TOSCC는 기밀성을 유지할 수 있는 기능을 가지게 된다. 또한, 트랜잭션에 대한 철회를 동일한 트랜잭션들 간에 충돌이 발생하는 것

으로 한정할 수 있게 됨에 따라 트랜잭션의 철회에 따른 가용성의 저하가 특정 보안등급을 부여 받은 트랜잭션군에 편중됨을 피할 수 있게 된다. 그 결과, 상이한 보안등급 트랜잭션의 비간섭적 수행을 위하여 특정 트랜잭션군들의 가용성이 저하되는 것을 방지할 수 있다. 따라서, TOSCC는 트랜잭션들간의 동시성 제어를 기밀성, 무결성, 가용성을 함께 획득하면서 수행할 수 있다.

V. 결 론

본 논문에서는 다중 등급 보안 데이터베이스 시스템상의 동시 수행 트랜잭션간의 동시성 제어를 하는 과정에서의 비밀경로를 제거하는 보안 동시성 제어기법 TOSCC를 제안하였다. TOSCC는 다중버전 데이터베이스를 기반으로 하고 있으며 트랜잭션간의 순서를 검증하여 불가피한 간섭이 발생하는 경우에만 트랜잭션을 철회시키고 있다. 따라서 직렬화가능성을 침해할 가능성만으로 철회되는 트랜잭션을 없앨 수 있게 되어 성능향상이 가능하다. 또한, 트랜잭션간의 간섭현상을 동일한 보안등급의 트랜잭션들간으로 한정함으로써 트랜잭션의 철회가 모든 보안등급에서 균등하게 발생한다. TOSCC는 다중버전 데이터베이스 시스템의 단점인 데이터베이스 대용량화를 무가치 데이터버전 삭제 기준을 제공하여 데이터베이스가 과도하게 증가하는 것을 방지할 수 있다.

본 논문에서는 제안한 기법에 대한 성능평가 결과를 제시하지 못하였다. 성능평가에 대한 추가적인 연구가 수행될 필요가 있다.

참고문헌

[1] C. Papadimitriou, The Theory of Database Concurrency Control, Computer Science Press, 1986.

[2] 김홍진, 오상엽, 김영선, "데이터베이스에서 다중 트랜잭션의 동시성 제어를 위한 직렬성 알고리즘 설계." 한국컴퓨터정보학회 논문집 6권 2호, 한국컴퓨터정보학회, 2001.

[3] Defense Information System Agency, "Multilevel Security in Department of Defense: The Basics," http://www.disa.mil/MLS/basics, 1995.

[4] T. F. Keefe, W. T. Tsai, and J. Srivastava, "Multilevel Secure Database Concurrency Control," Proc. Sixth International Conference on Data Engineering, Los Angeles, California, U.S.A, 1990.

[5] A. F. Mathur and .F. Keefe, "The Concurrency Control and Recovery Problem for Multilevel Update Transactions in M of Research in Security and Privacy," IEEE Computer Society Press, 1993.

[6] B. T. Blaustein, S. Jajodia, C. D. McCollum, and I. Motargiacomo, "A Model of Atomicity for Multilevel Transactions," Proceedings of Research in Security and Privacy, IEEE Computer Society Press, 1993.

[7] E. Bertino, S. Jajodia, L. Mancini, and I. Ray. "Advanced Transaction Processing in Multilevel Secure File Stones," To appear in IEEE Transactions on Knowledge and Data Engineering, 1997.

[8] J. McDermott and S. Jajodia, "Orange Locking; Channel-Free Database Concurrency Control Via Locking," Database Security, VI; Status and Prospects, Elsevier Science Publishers, 1993.

[9] K. P. Smith, B.T. Blaustein, and S. Jajodia, "Correctness Criteria for Multilevel Secure Transactions," IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 1, Feb, 1996.

[10] L. Asher and R. Hiltner, Trusted ORACLE7TM Server Administrator's Guide, Oracle Corproation, 1993.

[11] L. V. Mancini and I. Ray, "Secure Concurrency Control in MLS Databases with Two Versions of Data," Proceedings of European Symposium on Research in Computer Security, Springer Verlag, 1996.

[12] O. Costich and J. Dermott, "A Multilevel Transaction Problem for Multilevel Secure Database Systems and Its Solution for the Replicated Architecture," Proceedings of Reseach in Security and Privacy, IEEE Computer Society Press, 1992.

[13] O. Costich and S. Jajodia, "Maintaining Multilevel Transaction Atomicity in MSL Database Systems with Kernalized Architecture," Database Security, VI: Status and Prospects, Elsevier Science Publishers, 1993.

[14] P. Amman, F. Jaeckle, and S. Jajodia, "A Two Snapshot Algorithm for concurrency Control in Multi-Level Secure Databases," Proceedings of Research in Security and Privacy, IEEE Computer Society Press, 1992.

[15] P. Amman and S. Jajodia , "A Timestamp Ordering Algorithm for Secure, Single-Version, Multi-Level Databases," Database Security, V: Status and Prospects, Elsevier Science Publishers, 1992.

[16] S. Jajodia, L.V. Mancini, and I. Ray, "Secure Locking Protocols for Multilevel Database Management Systems," Database Security, X: Status and Prospects, Elsevier Science Publishers, 1996.

[17] S. Jajodia and V. Atluri, "Alternative Correctness Criteria for Concurrent Execution of Transactions in Multilevel Secure Databases," Proceedings of Research in Security and Privacy, IEEE Computer Society Press, 1992.

[18] T. F. Keefe and W. T. Tsai, "Multiversion Concurrency Control for Multilevel Secure Database Systems," Proceedings of Research in Security and Privacy, IEEE Computer Society Press, 1990.

[19] V. Atluri, S. Jajodia, and T.F. Keefe, "Multilevel Secure Transaction Processing: Status and Prospects," Database Socurity, X: Status and Prospects, Elsevier Science Publishers, 1996.

[20] V. Atluri, E. Bertino, and S. Jajodia, "Achieving Stricter Corretness Requirements in Multilevel Secure Databases," Proceedings of Reseach in Security and Privacy, IEEE Computer Society Press, 1933.

[21] V. Atluri, E. Bertino, and S. Jajodia, "Achieving Stricter Corretness Requirements in Multilevel Secure Databases: The Dynamic Case," Database Security, VII: Status and Prospects, Elsevier Science Publishers, 1994.

[22] W.T. Maimone and I.B. Greeberg, "Single-Level Multiversion Schedulers for Mulitilevel Secure Database Systems." Proceedings of 6th Computer Security Applications, IEEE Computer Society Press, 1990.

[23] C. P. Pfleeger, Security in Computing, Prentic e-Hall International, Inc. 1989.

[24] Canadian System Security Centre, The Canadian Trusted Computer Evaluation Criteria, Version 3.0e, 1993.

[25] D. Rushell and G.T. Gangemi Sr., Computer Security Basics. O'Retlly & Associates, Inc., 1991.

[26] National Computer Security Center, Trusted Computer System Evaluation Criteria, NCSC -TG-021, 1991.

[27] S. Castano, M. Fuguni, G. Martella, and P. Samarati, Database Security, Addison-Wesley, 1995.

저 자 소 개

이 원 섭

1999년 ~ 현재 인덕대학 컴퓨터전자전공 교수

<관심분야> 분산데이터베이스, 데이터베이스 설계

이 상 희

1996년~현재 청강문화산업대학 교수

<관심분야> 데이터베이스, 멀티데이터베이스, 모바일 시스템