(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0101232
(43) 공개일자 2012년09월13일

(51) 국제특허분류(Int. Cl.)
G06Q 50/00 (2006.01)
(21) 출원번호 10-2011-0017224
(22) 출원일자 2011년02월25일
심사청구일자 2011년02월25일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
정한재
경기도 수원시 팔달구 정자천로32번길 20, 꽃뫼버들마을 엘지아파트 165동 202호 (화서동)
원동호
서울특별시 서초구 잠원동 한일17차아파트 334동 903호
김승주
서울특별시 송파구 중대로 24, 올림픽훼밀리APT 234동 403호 (문정동)
(74) 대리인
특허법인 정안

전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **암호키를 갱신하여 게재글의 프라이버시를 보호할 수 있는 소셜 네트워크 서비스 시스템 및 방법**

(57) 요 약

다수의 사용자 단말과 네트워크로 연결되어, 사용자 사이에 관계가 형성되면 사용자가 게재한 글(이하 게재글)을 관계가 형성된 다른 사용자에게 열람시키는 소셜 네트워크 서비스 시스템 및 방법에 있어서, 사용자 단말로부터 사용자 가입이 요청되면, 사용자의 마스터 해시값으로 사용자의 현행키를 설정하는 가입관리부; 사용자 단말에 의해 사용자의 게재글이 작성되면, 사용자의 현행키로 게재글을 암호화하여 게재하되, 사용자의 현행키는 게재글의 게재순서에 따라 반복하여 일방향으로 갱신되는 글게재부; 사용자(이하 요청자)의 사용자 단말로부터 다른 사용자(이하 작성자)와의 관계형성이 요청되면, 작성자의 현행키 및 마지막 게재글의 게재순서를 요청자의 작성자에 대한 관계키 및 관계순서로 부여하는 관계형성부; 및, 요청자의 사용자 단말로부터 작성자의 게재글의 열람이 요청되면, 요청자의 작성자에 대한 관계키로 게재글을 복호화하여 사용자 단말로 전송하되, 관계키는 게재글의 게재순서와 관계순서와의 차이만큼 반복하여 일방향으로 갱신되는 글열람부를 포함하는 구성을 마련한다.

상기와 같은 시스템 및 방법에 의하여, 사용자가 작성한 모든 글을 암호화하고, 글을 작성할 때 마다 키를 업데이트함으로써, 관계가 형성된 이후의 글만 볼 수 있으므로 사용자의 프라이버시가 무분별하게 침해되는 것을 방지할 수 있다.

대 표 도 - 도3

표기법	설명
사용자A	사용자B와 관계를 형성하는 사용자
사용자B	사용자A에게 관계를 요청하는 사용자
r	랜덤한 수
H	암호학적으로 안전한 해시 함수
h_i	사용자i의 마스터 해시값
cur_h_i	사용자i의 현재 해시값
$count_i$	사용자i가 작성한 글 번호
$E_k(m)$	키 k를 이용하여 m을 암호화
$D_k(m)$	키 k를 이용하여 m을 복호화

특허청구의 범위

청구항 1

다수의 사용자 단말과 네트워크로 연결되어, 사용자 사이에 관계가 형성되면 사용자가 게재한 글(이하 게재글)을 관계가 형성된 다른 사용자에게 열람시키는 소셜 네트워크 서비스 시스템에 있어서,

상기 사용자 단말로부터 사용자 가입이 요청되면, 상기 사용자의 마스터 해시값으로 상기 사용자의 현행키를 설정하는 가입관리부;

상기 사용자 단말에 의해 사용자의 게재글이 작성되면, 상기 사용자의 현행키로 상기 게재글을 암호화하여 게재하되, 상기 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신되는 글게재부;

상기 사용자(이하 요청자)의 사용자 단말로부터 다른 사용자(이하 작성자)와의 관계형성이 요청되면, 상기 작성자의 현행키 및 마지막 게재글의 게재순서를 상기 요청자의 작성자에 대한 관계키 및 관계순서로 부여하는 관계형성부; 및,

상기 요청자의 사용자 단말로부터 상기 작성자의 게재글의 열람이 요청되면, 상기 요청자의 작성자에 대한 관계키로 상기 게재글을 복호화하여 상기 사용자 단말로 전송하되, 상기 관계키는 상기 게재글의 게재순서와 상기 관계순서와의 차이만큼 반복하여 일방향으로 갱신되는 글열람부를 포함하는 것을 특징으로 하는 소셜 네트워크 서비스 시스템.

청구항 2

제1항에 있어서,

상기 현행키 및 관계키는 해시함수에 의해 갱신되는 것을 특징으로 소셜 네트워크 서비스 시스템.

청구항 3

제1항에 있어서,

상기 관계형성부는 상기 요청자가 상기 작성자의 모든 게재글의 접근이 허용되면, 상기 작성자에 대한 관계키 및 관계순서를 상기 마스터 해시값 및 최초 게재글의 게재순서로 설정하는 것을 특징으로 소셜 네트워크 서비스 시스템.

청구항 4

제1항에 있어서,

상기 게재순서로서 카운터가 이용되되, 새로운 게재글이 하나씩 게재될 때마다 1씩 증가되는 것을 특징으로 소셜 네트워크 서비스 시스템.

청구항 5

제1항에 있어서,

상기 글열람부는 상기 작성자의 게재글의 열람이 요청되면, 상기 작성자의 마지막 게재글까지 열람시켜주고, 상기 요청자의 상기 작성자에 대한 관계키 및 관계순서를 상기 작성자의 현행키 및 마지막 게재글의 게재순서로 설정하는 것을 특징으로 소셜 네트워크 서비스 시스템.

청구항 6

서버 및 다수의 사용자 단말이 네트워크로 연결되어, 사용자 사이에 관계가 형성되면 사용자가 게재한 글(이하 게재글)을 관계가 형성된 다른 사용자에게 열람시키는 소셜 네트워크 서비스 방법에 있어서,

(a) 상기 사용자 단말이 사용자 가입을 요청하면, 상기 서버는 상기 사용자의 마스터 해시값으로 상기 사용자의 현행키를 설정하는 단계;

(b) 상기 사용자 단말에 의해 사용자의 게재글이 작성되면, 상기 서버는 상기 사용자의 현행키로 상기 게재글을 암호화하여 게재하되, 상기 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신되는 단계;

(c) 상기 사용자(이하 요청자)의 사용자 단말로부터 다른 사용자(이하 작성자)와의 관계형성이 요청되면, 상기 서버는 상기 작성자의 현행키 및 마지막 게재글의 게재순서를 상기 요청자의 작성자에 대한 관계키 및 관계순서로 부여하는 단계; 및,

(d) 상기 요청자의 사용자 단말로부터 상기 작성자의 게재글의 열람이 요청되면, 상기 서버는 상기 요청자의 작성자에 대한 관계키로 상기 게재글을 복호화하여 상기 사용자 단말로 전송하되, 상기 관계키는 상기 게재글의 게재순서와 상기 관계순서와의 차이만큼 반복하여 일방향으로 갱신되는 단계를 포함하는 것을 특징으로 하는 소셜 네트워크 서비스 방법.

청구항 7

제6항에 있어서,

상기 현행키 및 관계키는 해시함수에 의해 갱신되는 것을 특징으로 소셜 네트워크 서비스 방법.

청구항 8

제6항에 있어서,

상기 게재순서로서 카운터가 이용되되, 새로운 게재글이 하나씩 게재될 때마다 1씩 증가되는 것을 특징으로 소셜 네트워크 서비스 방법.

청구항 9

제6항에 있어서,

상기 (d)단계에서, 상기 작성자의 게재글의 열람이 요청되면, 상기 작성자의 마지막 게재글까지 열람시켜주고, 상기 요청자의 상기 작성자에 대한 관계키 및 관계순서를 상기 작성자의 현행키 및 마지막 게재글의 게재순서로 설정하는 것을 특징으로 소셜 네트워크 서비스 방법.

청구항 10

제1항 내지 제9항 중 어느 한 항의 방법을 수행하는 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

명 세 서

기 술 분 야

[0001] 본 발명은 사용자 사이에 관계가 형성되면 사용자가 게재한 글(이하 게재글)을 관계가 형성된 다른 사용자에게 열람시키는 소셜 네트워크 서비스 시스템 및 방법에 관한 것이다.

[0002] 특히, 본 발명은 작성자가 작성한 글을 암호화하되 암호키를 일방향으로 매번 갱신하고, 관계를 형성한 사용자에게 관계 형성시의 암호키를 공유시키는 소셜 네트워크 서비스 시스템 및 방법에 관한 것이다.

배 경 기 술

[0003] 최근 스마트폰의 사용이 급증함에 따라 트위터 또는 페이스북과 같은 소셜네트워크서비스(social network service, SNS)의 사용이 급증하고 있다. SNS는 뉴스 및 개인의 의견, 정보 등을 실시간으로 빠르게 공유할 수 있는 장점을 가지고 있다. 그러나 SNS를 통한 정보 공유가 개인의 사생활이나 과거에 작성되었던 글이 무분별하게 불특정 다수에게 공개되는 단점이 있다.

[0004] 페이스북은 API를 공개하여 웹의 사진, 동영상과 같은 정보뿐만 아니라 사용자들이 만드는 다양한 콘텐츠를 페이스북을 통하여 쉽게 공유할 수 있다. 페이스북에서 사용자 간 정보를 공유하기 위해서는 먼저 '친구' 라는 관계를 형성해야 한다. 페이스북의 특징 중 하나는 기본적으로 '친구' 관계를 맺은 사용자 간에만 정보가 공유되는 것이다. '친구' 관계를 맺기 위해서는 상대방에게 '친구'요청을 해야 하고 상대방 또한 수락을 해야 '친구' 관계가 성립된다. 사용자A와 사용자B가 친구관계라고 가정하면, 사용자A와 사용자B는 서로 작성한 글을 모두 볼 수 있다. 또한 사용자A가 글 또는 콘텐츠를 페이스북에 게시할 때 마다 사용자B는 사용자A가 글 또는 콘텐츠를 게시하였다는 것을 실시간으로 확인할 수 있다.

[0005] 페이스북은 이를 위해, 사용자 정보, 작성된 글, 사용자 간의 관계(또는 친구관계)에 대한 데이터를 서버에서 저장되어 관리된다. 즉, 페이스북에서는 사용자 정보를 아이디(id), 이름, 이메일, 거주지 등의 속성으로 서버에서 관리 및 저장되고 있다. 또한, 페이스북에서 작성된 글은 작성된 글의 아이디(id), 작성한 사용자의 아이디와 이름, 글 내용, 글을 작성할 때 사용한 어플리케이션 이름, 작성한 글에 가능한 행동 리스트(예를 들어, likes, 댓글 등), 작성된 글의 프라이버시 설정 등의 속성으로 서버에서 관리 및 저장되고 있다. 페이스북에서의 사용자 간 친구 관계를 맺게 되면, 친구관계인 사용자의 이름 및 아이디 등 속성으로 서버에서 관리되고 저장된다.

[0006] 트위터는 역시 페이스북과 같이 API를 공개하여 웹에 있는 다양한 정보를 트위터를 통해 공유할 수 있다. 트위터도 페이스북과 같이 사용자 간에 관계를 맺어 정보를 공유한다. 페이스북에서는 사용자 간에 '친구'라는 관계를 맺지만 트위터는 'Follow'라는 관계를 맺는다. 그러나 페이스북과 다르게 관계를 형성해야만 정보를 공유할 수 있는 것은 아니다. 트위터에서는 'Follow' 관계가 아닌 사용자도 특정 사용자를 검색하여 검색된 사용자가 작성한 글을 모두 볼 수 있다.

[0007] 트위터에서 'Follow' 관계는 임의의 사용자A가 트위터에 글을 작성하면, 사용자A를 'Follow'하고 있는 모든 사용자가 실시간으로 확인할 수 있는 관계이다. 또한 페이스북에서 관계를 형성하기 위해서는 승인이 필요하였지만, 트위터는 관계를 형성하기 위해서 승인이 필요없다. 트위터에서는 'Follow' 관계가 신청을 한 것과 받은 것에 따라 'Following'과 'Follower'로 구분된다.

[0008] 트위터에서는 사용자 정보를 사용자의 아이디 및 이름, 위치, 프로파일, 이메일 알람, 작성한 글의 수 등의 속성으로 서버에서 관리 및 저장되고 있다. 트위터에서 작성된 글은 글이 작성된 시간, 글의 아이디(id), 글이 작성된 프로그램 이름, 글 작성시의 위치 및 좌표, 작성된 글이 댓글인 경우 원본 글의 아이디 및 사용자 아이디 등 속성으로 서버에서 관리 및 저장되고 있다. 트위터에서 사용자 간 'follow' 관계를 맺게 되면 아이디 및 'follow'관계인 사용자의 아이디 등의 속성으로 서버에서 관리 및 저장되고 있다.

[0009] 페이스북과 트위터의 차이점을 정리하면 다음과 같다.

[0010] 작성한 글의 공유에 대한 기본 설정에 관해서, 페이스북은 친구에게만 공개하는데 반해, 트위터는 전체에게 공개된다. 업데이트 알림에 관해서, 페이스북은 친구에게만 알리는데 반해, 트위터는 'Follower'에게만 알린다. 단, 트위터에서, 설정을 통해 알림을 차단할 수 있다. 또한, 정보 차단 설정에 관하여, 페이스북은 친구 끊기를 통해 가능하나, 트위터는 불가능하다.

[0011] 한편, 페이스북이나 트위터에서 사용자B가 사용자A와 관계를 형성하였을 때, 사용자A가 작성한 글 중 사용자B가 읽을 수 있는 범위를 표현하면 도 1과 같다. 즉, 도 1과 같이, 일단 사용자 간에 서로 관계가 형성되면, 작성된 기간과 상관없이 모든 글(상대방이 작성한 모든 글)을 볼 수 있다.

[0012] 상기와 같은 종래의 소셜 네트워크 서비스에 의하면, 최근 사용자도 타 사용자(이하 작성자)와 관계를 형성하면, 바로 오래전 과거에 작성한 작성자의 글까지 모두 읽을 수 있다. 이를 이용하여, 악의의 사용자는 이를 악용할 소지가 많다.

[0013] 소셜 네트워크 서비스는 각 사용자가 즉시 현재에 느끼는 생각이나 느낌을 글로 표현하여 공유하는 것을 제공하는 서비스이다. 따라서 사용자는 시간에 따라 일관적이지 않은 글을 쓰거나, 그때 그때의 감정에 따라 서로 다른 생각의 글을 쓰기도 한다. 그런데, 악의의 사용자는 과거에 작성된 모든 작성자의 글들을 분석하여, 작성자에 불리한 글만 추려 공격하거나 모순된 글들만 대비하여 공격할 수도 있다. 또한, 악의의 사용자는 과거의 모든 작성자의 글을 통해, 작성자의 성향 등을 조사할 때 이용할 수도 있다.

[0014] 작성자와 오래도록 관계를 맺은 사용자들은 이러한 악의의 사용자는 드물다. 이것은 그만큼 관심있게 작성자의 글을 지속적으로 읽었던 사용자들은 선의의 사용자일 가능성이 높기 때문이다. 이에 반해, 악의의 사용자는 작성자와 즉시 관계를 맺은 뒤에 단시간 내에 작성자의 글을 분석하고 떠나는 경향이 많다.

[0015] 따라서 사용자가 작성자가 관계를 맺는 시기에 따라, 작성자가 작성한 글에 대한 접근을 제한할 필요가 있다. 그러나 종래의 소셜 네트워크 서비스는 이러한 기능을 제공해주고 있지 못하고 있다.

발명의 내용

해결하려는 과제

[0016] 본 발명의 목적은 상술한 바와 같은 문제점을 해결하기 위한 것으로, 사용자가 작성자와 관계를 맺으면, 관계를 맺은 이후에 작성된 글들만 사용자에게 열람시키는 소셜 네트워크 서비스 시스템 및 방법을 제공하는 것이다.

[0017] 또한, 본 발명의 목적은 작성자가 작성한 글을 암호화하되 암호키를 일방향으로 매번 갱신하고, 관계를 형성한 사용자에게 관계 형성시의 암호키를 공유시키는 소셜 네트워크 서비스 시스템 및 방법을 제공하는 것이다.

[0018] 또한, 본 발명의 목적은 관계를 형성한 사용자에게 특정 시점의 암호키를 공유시킴으로써 특정 시점 이후부터의 글을 열람시키는 소셜 네트워크 서비스 시스템 및 방법을 제공하는 것이다.

과제의 해결 수단

[0019] 상기 목적을 달성하기 위해 본 발명은 다수의 사용자 단말과 네트워크로 연결되어, 사용자 사이에 관계가 형성되면 사용자가 게재한 글(이하 게재글)을 관계가 형성된 다른 사용자에게 열람시키는 소셜 네트워크 서비스 시스템에 관한 것으로서, 상기 사용자 단말로부터 사용자 가입이 요청되면, 상기 사용자의 마스터 해시값으로 상기 사용자의 현행키를 설정하는 가입관리부; 상기 사용자 단말에 의해 사용자의 게재글이 작성되면, 상기 사용자의 현행키로 상기 게재글을 암호화하여 게재하되, 상기 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신되는 글게재부; 상기 사용자(이하 요청자)의 사용자 단말로부터 다른 사용자(이하 작성자)와의 관계형성이 요청되면, 상기 작성자의 현행키 및 마지막 게재글의 게재순서를 상기 요청자의 작성자에 대한 관계키 및 관계순서로 부여하는 관계형성부; 및, 상기 요청자의 사용자 단말로부터 상기 작성자의 게재글의 열람이 요청되면, 상기 요청자의 작성자에 대한 관계키로 상기 게재글을 복호화하여 상기 사용자 단말로 전송하되, 상기 관계키는 상기 게재글의 게재순서와 상기 관계순서와의 차이만큼 반복하여 일방향으로 갱신되는 글열람부를 포함하는 것을 특징으로 한다.

[0020] 또, 본 발명은 소셜 네트워크 서비스 시스템에 있어서, 상기 현행키 및 관계키는 해시함수에 의해 갱신되는 것을 특징으로 한다.

[0021] 또, 본 발명은 소셜 네트워크 서비스 시스템에 있어서, 상기 관계형성부는 상기 요청자가 상기 작성자의 모든 게재글의 접근이 허용되면, 상기 작성자에 대한 관계키 및 관계순서를 상기 마스터 해시값 및 최초 게재글의 게재순서로 설정하는 것을 특징으로 한다.

[0022] 또, 본 발명은 소셜 네트워크 서비스 시스템에 있어서, 상기 게재순서로서 카운터가 이용되되, 새로운 게재글이 하나씩 게재될 때마다 1씩 증가되는 것을 특징으로 한다.

[0023] 또, 본 발명은 소셜 네트워크 서비스 시스템에 있어서, 상기 글열람부는 상기 작성자의 게재글의 열람이 요청되면, 상기 작성자의 마지막 게재글까지 열람시켜주고, 상기 요청자의 상기 작성자에 대한 관계키 및 관계순서를 상기 작성자의 현행키 및 마지막 게재글의 게재순서로 설정하는 것을 특징으로 한다.

[0024] 또한, 본 발명은 서버 및 다수의 사용자 단말이 네트워크로 연결되어, 사용자 사이에 관계가 형성되면 사용자가 게재한 글(이하 게재글)을 관계가 형성된 다른 사용자에게 열람시키는 소셜 네트워크 서비스 방법에 관한 것으로서, (a) 상기 사용자 단말이 사용자 가입을 요청하면, 상기 서버는 상기 사용자의 마스터 해시값으로 상기 사용자의 현행키를 설정하는 단계; (b) 상기 사용자 단말에 의해 사용자의 게재글이 작성되면, 상기 서버는 상기 사용자의 현행키로 상기 게재글을 암호화하여 게재하되, 상기 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신되는 단계; (c) 상기 사용자(이하 요청자)의 사용자 단말로부터 다른 사용자(이하 작성자)와의 관계형성이 요청되면, 상기 서버는 상기 작성자의 현행키 및 마지막 게재글의 게재순서를 상기 요청자의 작성자에 대한 관계키 및 관계순서로 부여하는 단계; 및, (d) 상기 요청자의 사용자 단말로부터 상기 작성자의 게재글의 열람이 요청되면, 상기 서버는 상기 요청자의 작성자에 대한 관계키로 상기 게재글을 복호화하여 상기 사용자 단말로 전송하되, 상기 관계키는 상기 게재글의 게재순서와 상기 관계순서와의 차이만큼 반복하여 일방향으로 갱신되는 단계를 포함하는 것을 특징으로 한다.

[0025] 또, 본 발명은 소셜 네트워크 서비스 방법에 있어서, 상기 현행키 및 관계키는 해시함수에 의해 갱신되는 것을 특징으로 한다.

[0026] 또, 본 발명은 소셜 네트워크 서비스 방법에 있어서, 상기 게재순서로서 카운터가 이용되되, 새로운 게재글이 하나씩 게재될 때마다 1씩 증가되는 것을 특징으로 한다.

[0027] 또, 본 발명은 소셜 네트워크 서비스 방법에 있어서, 상기 (d)단계에서, 상기 작성자의 게재글의 열람이 요청되면, 상기 작성자의 마지막 게재글까지 열람시켜주고, 상기 요청자의 상기 작성자에 대한 관계키 및 관계순서를 상기 작성자의 현행키 및 마지막 게재글의 게재순서로 설정하는 것을 특징으로 한다.

[0028] 또한, 본 발명은 상기 방법을 수행하는 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것이다.

[0029]

발명의 효과

[0030] 상술한 바와 같이, 본 발명에 따른 소셜 네트워크 서비스 시스템 및 방법에 의하면, 사용자가 작성한 모든 글을 암호화하고, 글을 작성할 때 마다 키를 업데이트함으로써, 관계가 형성된 이후의 글만 볼 수 있게 함으로써, 사용자의 프라이버시가 무분별하게 침해되는 것을 방지할 수 있는 효과가 얻어진다.

도면의 간단한 설명

[0031] 도 1은 종래의 소셜 네트워크 서비스에 따라 사용자가 열람할 수 있는 작성자의 글의 범위를 표시한 도표이다.

도 2는 본 발명을 실시하기 위한 전체 소셜 네트워크 서비스 구성의 일례를 도시한 도면이다.

도 3은 본 발명의 설명을 위한 표기법을 도시한 표이다.

도 4는 본 발명의 일실시예에 따른 소셜 네트워크 서비스 시스템의 구성에 대한 블록도이다.

도 5는 본 발명의 일실시예에 따라 사용자가 가입될 때의 추가되는 정보를 도시한 표이다.

도 6은 본 발명의 일실시예에 따른 현행키 또는 관계키의 갱신 알고리즘을 도시한 것이다.

도 7은 본 발명의 일실시예에 따라 현행키가 갱신되어 게재글을 복호화하는 것을 도시한 것이다.

도 8은 본 발명의 일실시예에 따라 관계가 형성될 때의 추가되는 정보를 도시한 표이다.

도 9는 본 발명의 일실시예에 따라 게재글 열람하는 방법을 설명하는 흐름도이다.

도 10은 본 발명의 일실시예에 따른 소셜 네트워크 서비스 방법을 설명하는 흐름도이다.

도 11은 본 발명과 종래 기술의 보호범위를 대비한 도면이다.

도 12는 본 발명에 따른 실험결과에 의한 추가 처리 시간을 기록한 표이다.

* 도면의 주요 부분에 대한 부호의 설명 *

10 : 사용자 단말 11 : 작성자 단말

12 : 요청자 단말 20 : 인터넷

30 : 서버 31 : 가입관리부

32 : 글게재부 33 : 관계형성부

34 : 글열람부 40 : 데이터베이스

41 : 사용자DB 42 : 게재글DB

43 : 관계DB

발명을 실시하기 위한 구체적인 내용

[0032] 이하, 본 발명의 실시를 위한 구체적인 내용을 도면에 따라서 설명한다.

[0033] 또한, 본 발명을 설명하는데 있어서 동일 부분은 동일 부호를 붙이고, 그 반복 설명은 생략한다.

[0034] 먼저, 본 발명을 실시하기 위한 전체 소셜 네트워크 서비스 구성의 일례를 도 2를 참조하여 설명한다.

[0035] 도 2에서 도시한 바와 같이, 본 발명을 실시하기 위한 전체 소셜 네트워크 서비스 구성은 사용자 단말(10), 네트워크(20), 및 서버(30)로 구성된다.

[0036] 사용자 단말(10)은 글을 작성하는 작성자 또는 글 열람을 요청하는 요청자 등 사용자가 이용하는 PC, 노트북, 넷북, PDA, 스마트폰 등의 통상의 컴퓨팅 단말기이다. 사용자는 사용자 단말(10)을 이용하여, 글을 작성하여 게재하거나, 작성된 글을 요청하여 열람한다.

[0037] 특히, 사용자 단말(10)은 작성자가 사용하는 작성자 단말(11)과 글 열람을 요청하는 요청자 단말(12)로 구분할 수 있다. 그러나 작성자 단말(11)과 요청자 단말(12)은 동일한 사용자 단말(10)을 기능적으로 구분한 것이다. 즉, 동일한 사용자 단말(10)이 글을 작성할 때 이용되면 작성자 단말(11)이 되고, 글 열람에 이용되면 요청자 단말(12)이 된다.

[0038] 한편, 사용자는 소셜 네트워크 서비스와 관련하여 어떤 작업을 수행할 때, 항상 사용자 단말(10)을 이용하여 작업을 수행한다. 따라서 이하에서, 사용자가 어떤 작업을 수행한다는 기술은 곧 사용자 단말(10)을 통해 작업을 수행함을 의미한다. 이런 의미에서, 도면 부호 10도 사용자 단말(10)뿐만 아니라 사용자(10)에게도 붙이기로 한다. 동일한 의미에서, 작성자 단말(11) 및 요청자 단말(12)의 도면부호도 작성자(11) 및 요청자(12)에도 붙이기로 한다.

[0039] 서버(30)는 소셜 네트워크 서비스를 제공하는 통상의 서버로서, 네트워크(20)에 연결되어, 사용자(10)의 요청에 따라 가입을 시키고, 사용자의 요청에 따라 사용자 간의 관계를 형성시켜준다. 또한, 서버(30)는 작성자(11)가 글을 작성하여 게재하면, 그 글을 저장한다. 그리고 서버(30)는 작성자(11)와 관계를 맺은 요청자(12)의 열람 요청에 의해, 그 글을 열람시켜줄 수 있게 한다.

[0040] 한편, 작성자(11)는 각자가 자신만의 암호키를 보유하여, 그 암호키로 작성한 글을 암호화한다. 즉, 서버(30)는 작성자(11)의 암호키에 의해 암호화된 글(이하 게재글)을 게재한다.

[0041] 그리고 작성자(11)는 자신과 관계를 맺은 요청자(12)에게 암호키를 공유시킨다. 요청자(12)는 암호키로 작성자(11)의 암호화된 게재글을 복호화하여 열람한다. 따라서 작성자(11)의 글은 작성자(11)와 관계를 맺은 요청자(12)만이 복호화하여 읽을 수 있다.

[0042] 데이터베이스(40)는 사용자 정보를 저장하는 사용자DB(41), 게재된 글에 대한 정보를 저장하는 게재글DB(42), 사용자 간의 관계의 정보를 저장하는 관계DB(43)로 이루어진다. 그러나 상기 데이터베이스(40)의 구성은 바람직한 일실시예일 뿐이며, 구체적인 장치를 개발하는데 있어서, 접근 및 검색의 용이성 및 효율성 등을 감안하여 데이터베이스 구축이론에 의하여 다른 구조로 구성될 수 있다.

[0043] 다음으로, 이후에 설명될 표기법을 도 3을 참조하여 설명한다.

[0044] 도 3에서 보는 바와 같이, 사용자A와 사용자B는 사용자의 일례로서, 사용자A는 사용자B와 관계를 형성하는 사용자 또는 작성자(11)이고, 사용자B는 사용자A에게 관계를 요청하는 사용자 또는 요청자(12)이다.

[0045] 또한, 해시함수 H는 일방향 함수이다. $count_i$ 는 사용자 i가 작성한 글 번호(또는 게재글의 게재순서)이다.

[0046] 다음으로, 본 발명의 제1 실시예에 따른 소셜 네트워크 서비스 시스템의 구성을 도 4를 참조하여 보다 구체적으로 설명한다.

[0047] 도 4에서 보는 바와 같이, 본 발명에 따른 서버(30) 또는 소셜 네트워크 서비스 시스템(30)은 가입관리부(31), 글게재부(32), 관계형성부(33), 글열람부(34)를 포함하여 구성된다.

[0048] 가입관리부(31)는 사용자 단말(10)로부터 사용자 가입이 요청되면, 상기 사용자(10)의 마스터 해시값으로 사용자의 현행키를 설정한다.

[0049] 도 5에서 보는 바와 같이, 사용자A가 서버(30)에 가입 신청을 하게 되면, 가입관리부(31)는 기존에 서버(30)에 저장되는 사용자A의 정보와 함께, 마스터 해시값 h_A, 현행키 cur_h_A , 카운터 $count_A$ 를 추가로 저장한다.

[0050] 마스터 해시값 h_A는 랜덤한 수 r을 해시함수 H로 해시한 값으로서, 사용자A에 고유한 값으로 설정된다. 현행키 cur_h_A 는 작성한 글을 암호화할 때 사용하는 암호키로서, 작성한 글(또는 게재글)이 게재되는 순서(또는 게재순서)에 의해 매번 갱신된다. 카운터 $count_A$ 는 사용자 A(또는 작성자)가 작성한 글의 일련 번호(또는 게재순서)를 나타낸다.

[0051] 글게재부(32)는 상기 사용자 단말에 의해 사용자의 게재글이 작성되면, 상기 사용자의 현행키로 상기 게재글을 암호화하여 게재한다. 이때, 상기 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신된다.

[0052] 사용자A가 새로운 글 m을 작성하면, 게재글 m을 그대로 서버(30)에 저장하지 않고 현행키 cur_h_A를 암호키로 사용하여 m을 암호화하여 저장한다. 그리고 추가로 현재 게재글 m이 사용자A가 몇 번째 작성한 글(또는 글의 게재순서)인지 표시하는 카운터 $count_A$를 함께 저장한다. 즉, 게재순서로서 카운터가 이용되되, 새로운 게재글이 하나씩 게재될 때마다 1씩 증가된다.

[0053] 마지막으로 게재글 m이 저장된 후, 글게재부(32)는 사용자A의 현행키 cur_h_A와 카운터 $count_A$를 도 6과 같은 알고리즘을 이용하여 업데이트 한다.

[0054] 도 7에서 보는 바와 같이, 게재글 M1, M2, ..., M6이 작성되어 게재되는 경우, 각 게재글은 현행키 $cur\_h_A^1$, $cur\_h_A^2$, ..., $cur\_h_A^6$ 에 의해 암호화된다. 이때, $cur\_h_A^1$ 를 해시하여 $cur\_h_A^2$ (= $H(cur\_h_A^1)$)를 생성하고, 다시 $cur\_h_A^2$ 를 해시하여 $cur\_h_A^3$ (= $H(cur\_h_A^2)$)을 생성한다. 이와 같이, 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신된다.

[0055] 관계형성부(33)는 사용자(이하 요청자)의 사용자 단말(12)로부터 다른 사용자(이하 작성자)(11)와의 관계형성이 요청되면, 상기 작성자의 현행키 및 마지막 게재글의 게재순서를 상기 요청자의 작성자에 대한 관계키 및 관계

순서로 부여한다.

[0056] 즉, 사용자B(또는 요청자)가 사용자A(또는 작성자)와 관계를 형성하였을 때, 관계형성부(33)는 사용자A에 대하여 도 8과 같은 정보를 추가로 저장한다.

[0057] 예를 들어, 도 7의 예에서, 게재글 M3까지 작성이 완료되고, 아직 게재글 M4 ~ M6이 작성되지 않는 경우를 설명한다. 이때, 요청자(또는 사용자B)의 마스터 해시값(또는 마스터 관계키) $master\_h_{BA}$ 을 cur_h_{BA} 로 설정한다. 그리고 관계키 cur_h_{BA} 도 $cur\_h_A^3$ 으로 설정된다.

[0058] 즉, 사용자B(요청자)의 사용자A(작성자)에 대한 마스터 관계키 $master\_h_{BA}$ 및 관계키 cur_h_{BA} 는 사용자B(요청자)가 사용자A(작성자)의 게재글을 열람하기 위한 암호키들을 말한다. 관계키 cur_h_{BA} 는 요청자가 작성자의 게재글을 읽을 때마다 해시함수를 이용하여 갱신된다. 즉, 도 7의 예에서, 처음 관계키 cur_h_{BA} 는 $cur\_h_A^3$ 으로 설정되어, 작성자의 글 M3을 복호화하여 읽을 수 있고, 상기 관계키를 해시함수로 해시하여, cur_h_{BA} 는 $cur\_h_A^4$ 로 변경된다. 그리고 작성자의 다음 글 M4를 복호화하여 열람한다. 이렇게 관계키를 반복하여 일방향으로 갱신하여, 순차적으로 작성자(사용자A)의 글 M3, M4, M5, M6, ... 을 복호화하여 읽을 수 있다.

[0059] 한편, 해시함수는 일방향 함수이므로, 사용자B(요청자)는 $cur\_h_A^3$ 으로 $cur\_h_A^2$, $cur\_h_A^1$ 을 계산할 수 없다. 따라서 요청자는 작성자와 관계를 맺기 전의 작성된 글인 M1, M2를 읽을 수 없다.

[0060] 또한, 관계 마스터 카운터(또는 관계순서) $master\_count_{BA}$ 와 관계 카운터 $count_{BA}$ 는 작성자(사용자A)와 관계를 형성할 때의 작성자가 마지막에 작성한 글의 카운터 값으로 설정한다. 그 후, 관계 카운터 $count_{BA}$ 는 요청자가 작성자의 글을 읽으면서 하나씩 증가시키는 카운터다. 즉, 관계 카운터는 작성자의 글 중 요청자가 읽은 마지막 글의 일련번호(또는 게재순서)를 가리킨다.

[0061] 글열람부(34)는 요청자(12)의 사용자 단말로부터 상기 작성자(12)의 게재글의 열람이 요청되면, 상기 요청자의 작성자에 대한 관계키로 상기 게재글을 복호화하여 사용자 단말로 전송한다. 이때, 상기 관계키는 상기 게재글의 게재순서와 상기 관계순서와의 차이만큼 반복하여 일방향으로 갱신된다.

[0062] 앞서 설명한 바와 같이, 요청자는 작성자의 관계키로 작성자의 글을 복호화하여 열람한다. 이때, 관계키를 해시함수로 해시하여 갱신하면, 갱신된 관계키로 바로 다음의 게재글을 열람할 수 있다. 도 7의 예로 설명하면, 작성자의 6번째 게재글 M6은 암호키 $cur\_h_A^6$ 를 알아야 열람할 수 있다. 요청자가 처음 관계를 맺으면서 공유한 관계키 $cur\_h_A^3$ 이므로, $cur\_h_A^3$ 을 3번 반복하여 해시하면 $cur\_h_A^6$ 를 구할 수 있다. 즉, 3번은 읽고자 하는 게재글의 게재순서 6과 관계를 맺을 때의 관계순서 3 과의 차이이다.

[0063] 일반적으로, 요청자(사용자B)가 작성자의 글을 요청하면, 이전에 읽었던 글부터 현재까지 작성된 게재글을 모두 열람시켜준다. 따라서 글열람부(34)는 상기 작성자의 게재글의 열람이 요청되면, 상기 작성자의 마지막 게재글까지 열람시켜주고, 결과적으로 요청자의 상기 작성자에 대한 관계키 및 관계순서를 상기 작성자의 현행키 및 마지막 게재글의 게재순서로 설정한다.

[0064] 다음으로, 글열람부(34)에 의해, 카운터를 이용하여 작성자(사용자A)의 게재글을 열람하는 과정을 도 6을 참조하여 보다 구체적으로 살펴본다.

[0065] 사용자B가 관계를 형성한 사용자A의 글을 서버에 요청하였을 때, 글열람부(34)는 다음과 같은 과정을 통해 사용자B에게 결과를 전송한다.

[0066] 단계 1: $count_{BA} > count_A$

[0067] $count_{BA}$가 m의 $count_A$보다 크다는 것은 이전에 작성된 글을 읽으려는 것을 의미한다. 이러한 경우 우선 사용자B와 A의 관계 형성 시점을 확인하기 위해 $master_count_{BA}$와 $count_A$를 한 번 더 비교한다. $master_count_{BA}$가

$count_A$보다 크다는 것은 사용자B가 사용자A와 관계를 형성하기 이전의 글을 읽으려고 한 것이므로 사용자B의 요청을 거부하고 'fail'을 반환한다. $master_count_{BA}$가 $count_A$보다 크지 않은 것은 현재의 글은 아니지만 관계 형성 이후의 글을 읽으려고 하는 것이므로 $count_{BA}=master\_count_{BA}$, $cur_h_{BA}=master_h_{BA}$로 설정하고 단계 2로 넘어간다.

[0068] 단계 2: $count_{BA} = count_A$

[0069] $count_{BA}$가 m의 $count_A$와 같은 것은 올바른 요청이므로 cur_h_{BA}를 키로 사용하여 m을 복호화하여 m'을 얻는다. 그 후 $Alg.update(cur_h_{BA}, count_{BA})$를 수행하여, cur_h_{BA}와 $count_{BA}$를 업데이트 한 후 m'을 반환한다.

[0070] 단계 3: $count_{BA} <= count_A$

[0071] $count_{BA}$가 m의 $count_A$보다 작거나 같은 것은 올바른 요청이다. 그러나 키가 업데이트가 되지 않은 상태이므로 $count_{BA}$와 $count_A$가 같을 때 까지 $Alg.update(cur_h_{BA}, count_{BA})$를 수행한다. 그 후, cur_h_{BA}를 키로 사용하여 m을 복호화하여 m'을 얻는다. 마지막으로 $Alg.update(cur_h_{BA}, count_{BA})$를 한 번 더 수행하여, cur_h_{BA}와 $count_{BA}$를 업데이트 한 후 m'을 반환한다.

[0072] 다음으로, 본 발명의 제2 실시예에 따른 소셜 네트워크 서비스 시스템의 구성을 설명한다. 본 발명의 제2 실시예는 앞서 제1 실시예의 변형으로서, 앞서 설명한 부분과 다른 부분만 설명한다. 본 발명의 제2 실시예는 요청자에게 작성자의 모든 글을 볼 수 있게 하는 실시예이다.

[0073] 관계형성부(33)는 요청자가 상기 작성자의 모든 게재글의 접근이 허용되면, 상기 작성자에 대한 관계키 및 관계순서를 상기 마스터 해시값 및 최초 게재글의 게재순서로 설정한다.

[0074] 즉, 사용자A는 사용자B에게 관계 형성 이전의 글을 읽을 수 있도록 허가할 수 있다. 사용자A가 사용자B에게 자신이 작성한 모든 글의 열람을 허용할 경우, 관계형성부(33)는 다음 [수학식 1]과 같이 값을 변경한다.

[0075] [수학식 1]

[0076] $master\_h_{BA} = cur\_h_{BA} = H(h_A)$

[0077] $master\_count_{BA} = count_{BA} = 1$

[0078] 이와 같이 설정하면, 사용자B는 사용자A가 작성한 모든 글을 열람할 수 있는 권한을 얻게 된다. 이것은 요청자(사용자B)가 작성자(사용자A)의 마스터 해시값을 가질 수 있기 때문에, 작성자의 모든 글을 읽을 수 있다.

[0079] 도 7의 예에서, 요청자는 첫 번째 글의 암호키이자 마스터 해시값인 $cur\_h_A^1$ 을 부여받는다. 따라서 작성자의 글 M1 ~ M6 을 모두 열람할 수 있다.

[0080] 다음으로, 본 발명의 일실시예에 따른 소셜 네트워크 서비스 방법을 도 10을 참조하여 설명한다. 상기 방법은 서버(30) 및 다수의 사용자 단말(10)이 네트워크로 연결된 소셜 네트워크 서비스 시스템에서 수행된다.

[0081] 도 10에서 보는 바와 같이, 소셜 네트워크 서비스 방법은 (a) 사용자 단말(10)이 사용자 가입을 요청하면, 서버(30)는 상기 사용자의 마스터 해시값으로 상기 사용자의 현행키를 설정하는 단계(S10); (b) 상기 사용자 단말(10)에 의해 사용자의 게재글이 작성되면, 상기 서버(30)는 상기 사용자의 현행키로 상기 게재글을 암호화하여 게재하되, 상기 사용자의 현행키는 상기 게재글의 게재순서에 따라 반복하여 일방향으로 갱신되는 단계(S20); (c) 상기 사용자(이하 요청자)의 사용자 단말(10)로부터 다른 사용자(이하 작성자)와의 관계형성이 요청되면, 상기 서버(30)는 상기 작성자의 현행키 및 마지막 게재글의 게재순서를 상기 요청자의 작성자에 대한 관계키 및 관계순서로 부여하는 단계(S30); 및, (d) 상기 요청자의 사용자 단말(10)로부터 상기 작성자의 게재글의 열람이

요청되면, 상기 서버(30)는 상기 요청자의 작성자에 대한 관계키로 상기 게재글을 복호화하여 상기 사용자 단말로 전송하되, 상기 관계키는 상기 게재글의 게재순서와 상기 관계순서와의 차이만큼 반복하여 일방향으로 갱신되는 단계(S40)로 구분된다.

[0082] 특히, 현행키 및 관계키는 해시함수에 의해 갱신된다. 또한, 상기 게재순서로서 카운터가 이용되되, 새로운 게재글이 하나씩 게재될 때마다 1씩 증가된다.

[0083] 또한, 상기 (d)단계에서, 상기 작성자의 게재글의 열람이 요청되면, 상기 작성자의 마지막 게재글까지 열람시켜주고, 상기 요청자의 상기 작성자에 대한 관계키 및 관계순서를 상기 작성자의 현행키 및 마지막 게재글의 게재순서로 설정한다.

[0084] 다음으로, 본 발명과 종래기술과의 보호수준에 대한 대비를 도 11을 참조하여 설명한다.

[0085] 트위터는 사용자 간의 관계 형성 없이도 사용자가 작성된 모든 글이 불특정 다수에게 노출될 수 있다. 즉, 사용자의 프라이버시 보호가 전혀 이루어지지 않는다고 할 수 있다.

[0086] 페이스북은 트위터보다 한 단계 수준 높은 프라이버시를 제공하고 있다. 페이스북은 사용자 간에 관계를 형성한 후에 정보를 공유할 수 있다. 그러나 한 번 관계를 형성하면 관계를 맺기 이전의 글까지 모두 공유될 수 있는 문제점이 있다.

[0087] 본 발명은 사용자B가 사용자A와 관계를 형성하였을 때, 사용자B는 관계 형성 이후에 사용자A가 작성한 글만 열람할 수 있도록 제한하였다. 따라서 관계 형성 이전에 작성된 글이 공유됨에 따라 침해받을 수 있는 프라이버시를 보호할 수 있다.

[0088] 도 10은 본 발명과 페이스북, 트위터에서 사용자B와 사용자A가 관계를 형성했을 때, 사용자A가 작성한 글을 사용자B가 읽을 수 있는 범위를 비교한 것이다. 또한 프라이버시 보호 수준에 따라 단계를 나누어서 표기하였다.

[0089]

[0090] 다음으로, 본 발명을 기존 SNS(소셜 네트워크 서비스)에 적용하였을 경우 요구되는 추가 공간 및 연산 시간에 대한 분석을 도 12를 참조하여 설명한다.

[0091] (1) 추가 공간 분석

[0092] 사용자 정보에 추가되는 데이터는 h_A, cur_h_A이다. h_A, cur_h_A는 해시 값으로 현재, 암호학적으로 안전하다고 알려진 해시함수의 길이는 256비트이다. 그리고 $count_A$는 글을 작성할 때 필요한 일련번호이다. 1초에 글을 하나 작성하고, 인간의 수명이 80년이라고 가정하였을 때, count는 최대 2,522,880,000까지 증가할 수 있다. 이를 비트로 표현하면 32비트이므로 $count_A$는 32비트면 충분하다. 따라서 사용자 정보에 추가로 필요한 공간은 544비트, 즉 68바이트이다.

[0093] 그리고 사용자 간에 관계를 형성할 때, 추가되는 데이터는 $master\_h_{BA}$, cur_h_{BA}, $master\_count_{BA}$, $count_{BA}$이다. $master\_h_{BA}$, cur_h_{BA}는 해시값으로 현재 암호학적으로 안전하다고 알려진 해시함수의 길이는 256비트이다. 그리고 $master_count_{BA}$와 $count_{BA}$는 글을 작성할 때 필요한 일련번호이며, 32비트면 충분하다. 따라서 사용자 간에 관계를 형성할 때, 추가로 필요한 공간은 576비트, 즉 72바이트이다.

[0094] 본 발명을 적용함으로써, 추가로 필요한 공간을 정리하면 다음과 같다.

[0095] (256bit ㅅ 2 + 32bit) + (256bit ㅅ 2 + 32bit ㅅ 2) ㅅ n

[0096] (단, n은 관계를 형성한 사용자 수)

[0097] (2) 추가 처리 시간 분석

[0098] 본 발명을 적용하였을 때, 추가로 필요한 처리 시간을 분석하기 위해 현재까지 안전하다고 알려진 알고리즘을 선택하여 분석을 하였다. 해시함수는 SHA-256을 사용하고, 암호 알고리즘은 128비트의 키를 사용하여 AES-CBC를 이용하는 것으로 가정하고 분석을 수행하였다. AES-CBC에 입력으로 들어가는 256비트의 키는 0~127비트는 IV로

사용되며, 128~255비트는 키로 사용된다.

[0099] 본 발명을 적용하였을 때, 가입, 글 작성, 글 읽기에 추가로 필요한 해시, 암/복호화 수를 정리하면 도 12a와 같다. 단, 글 읽기는 "단계 2"로 가정한다.

[0100] crypto++ 라이브러리를 Intel Core 2 1.83 GHz 프로세서, 32비트 Windows 비스타에서 구동했을 때, SHA-256은 0.275microsecond, AES-CBC는 1.225microsecond, AES-CBC의 IV 및 키 설정 시간이 0.569microsecond이다. 단, SNS에서 작성되는 글의 길이는 다양한데, 트위터의 경우 기본적으로 글의 길이를 140바이트로 제한하므로 본 논문에서도 글의 길이를 140바이트로 가정하고 암/복호화했을 때의 시간을 계산하였다. 본 발명을 적용했을 때 추가로 소요되는 시간을 정리하면 도 12b와 같다.

[0101] 마지막으로, 본 발명의 보안분석을 설명한다.

[0102] 보안은 Confidentiality, Key freshness, Backward secrecy의 3가지 측면에서 살펴본다.

[0103] 본 발명은 SNS에서 글을 작성할 때마다 128비트의 키를 이용하여 AES-CBC로 암호화하므로 confidentiality를 제공한다.

[0104] 본 발명은 글을 작성할 때마다, 암호학적으로 안전한 해시함수를 이용하여 글을 암호화하는 키를 갱신하므로 key freshness를 만족한다.

[0105] 사용자 B가 사용자A와 관계를 형성하면 사용자 A의 글을 복호화할 때 키로 사용할 cur_h_{BA}를 갖게 된다. cur_h_{BA}는 글을 작성할 때 마다 $cur_h_{BA}=H(cur_h_{BA})$와 같이 갱신 된다. 사용자B가 cur_h_{BA}로부터 이전의 cur_h_{BA}를 계산할 수 있으면 H의 pre-image를 찾은 것과 같다. 그러나 H는 암호학적으로 안전한 해시 함수라고 가정하였으므로 cur_h_{BA}를 이용하여 이전의 cur_h_{BA}를 찾을 수는 없다. 따라서 본 발명은 backward secrecy를 만족한다.

[0106] 이상, 본 발명자에 의해서 이루어진 발명을 실시 예에 따라 구체적으로 설명하였지만, 본 발명은 실시 예에 한정되는 것은 아니고, 그 요지를 이탈하지 않는 범위에서 여러 가지로 변경 가능한 것은 물론이다.

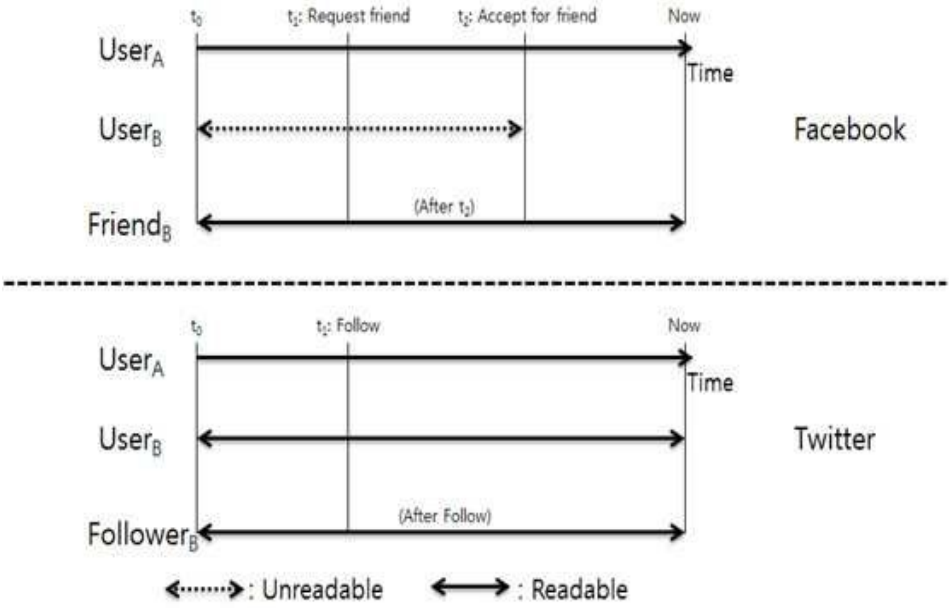
[0107]

산업상 이용가능성

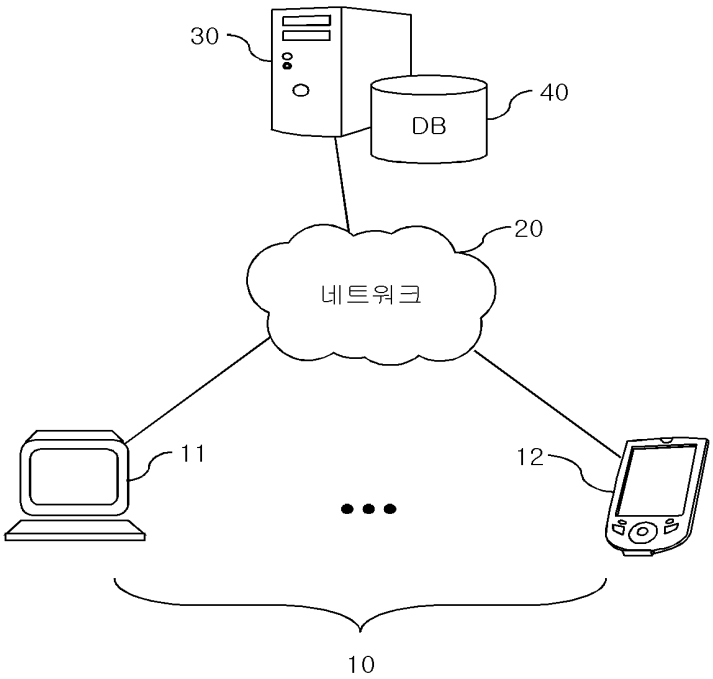
[0108] 본 발명은 사용자가 작성자와 관계를 맺으면, 관계를 맺은 이후에 작성된 글들만 사용자에게 열람시키는 소셜 네트워크 서비스 시스템을 개발하는 데 적용이 가능하다.

도면

도면1


t0
t1: Request friend
t2: Accept for friend
Now
User_A
Time
User_B
Facebook
Friend_B
(After t2)
t0
t1: Follow
Now
User_A
Time
User_B
Twitter
Follower_B
(After Follow)
: Unreadable
: Readable

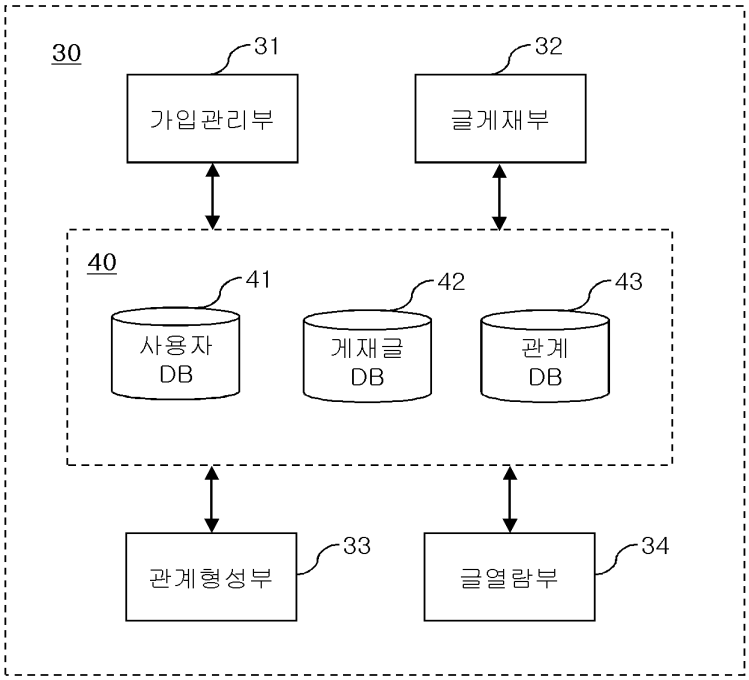
도면2


30
DB
40
20
네트워크
11
12
10

도면3

표기법	설명
사용자A	사용자B와 관계를 형성하는 사용자
사용자B	사용자A에게 관계를 요청하는 사용자
r	랜덤한 수
H	암호학적으로 안전한 해시 함수
h_i	사용자i의 마스터 해시값
cur_h_i	사용자i의 현재 해시값
$count_i$	사용자i가 작성한 글 번호
$E_k(m)$	키 k를 이용하여 m을 암호화
$D_k(m)$	키 k를 이용하여 m을 복호화

도면4



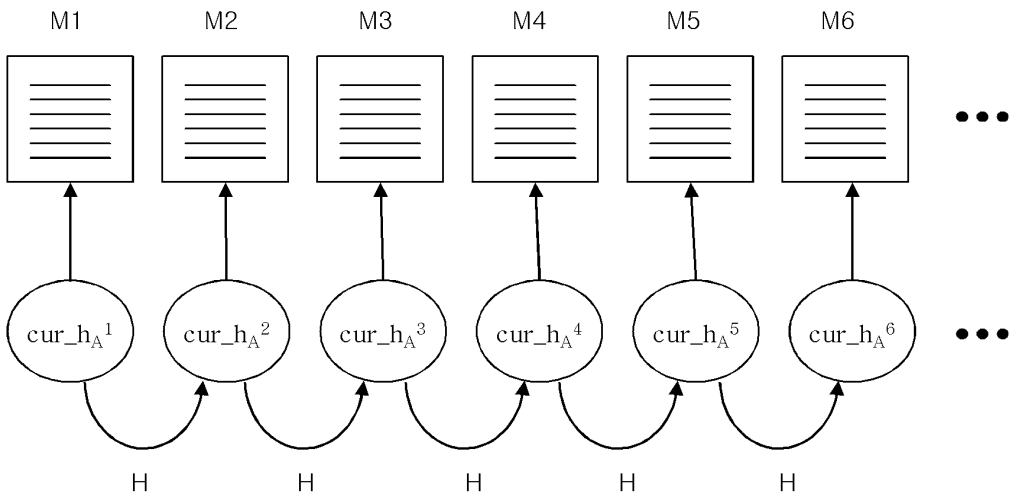
도면5

추가로 저장되는 정보	데이터	설명
h_A	=H(r)	마스터 해시값
cur_h_A	=H(h)	작성한 글을 암호화 할 때 사용되는 키이며, 글을 작성할 때마다 해시 함수를 이용하여 업데이트 됨
$count_A$	=1	사용자가 작성한 글의 일련 번호

도면6

```
Alg.Update(cur_h, count) {
    cur_h = H(cur_h);
    count = count + 1;
    return cur_h, count;
}
```

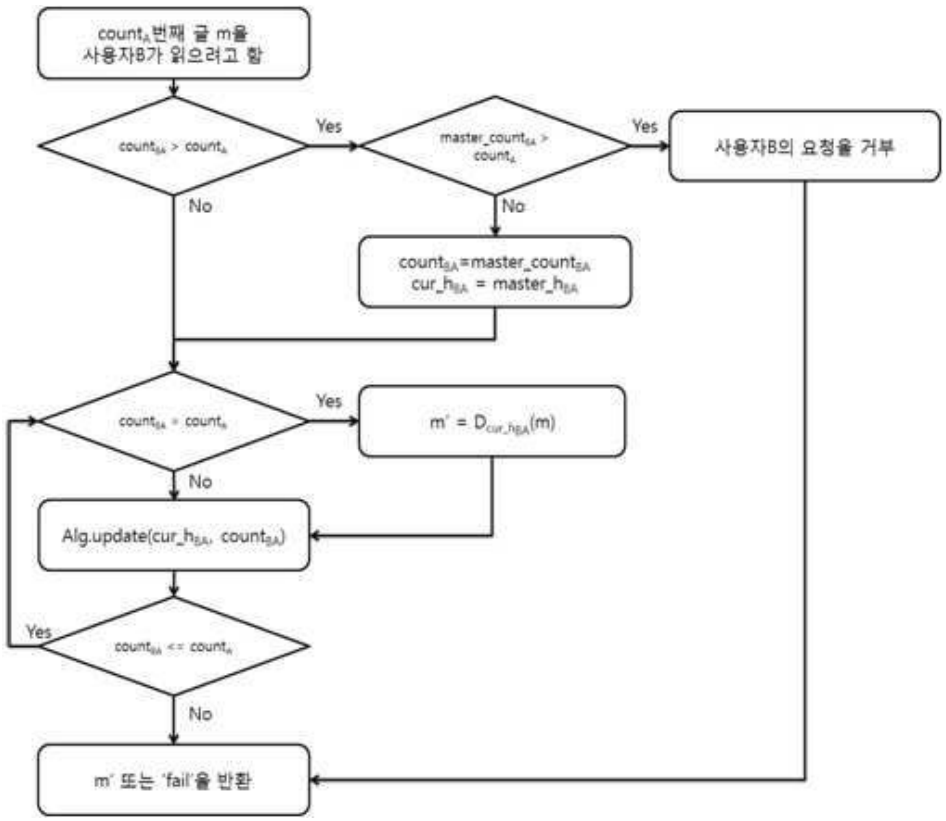
도면7



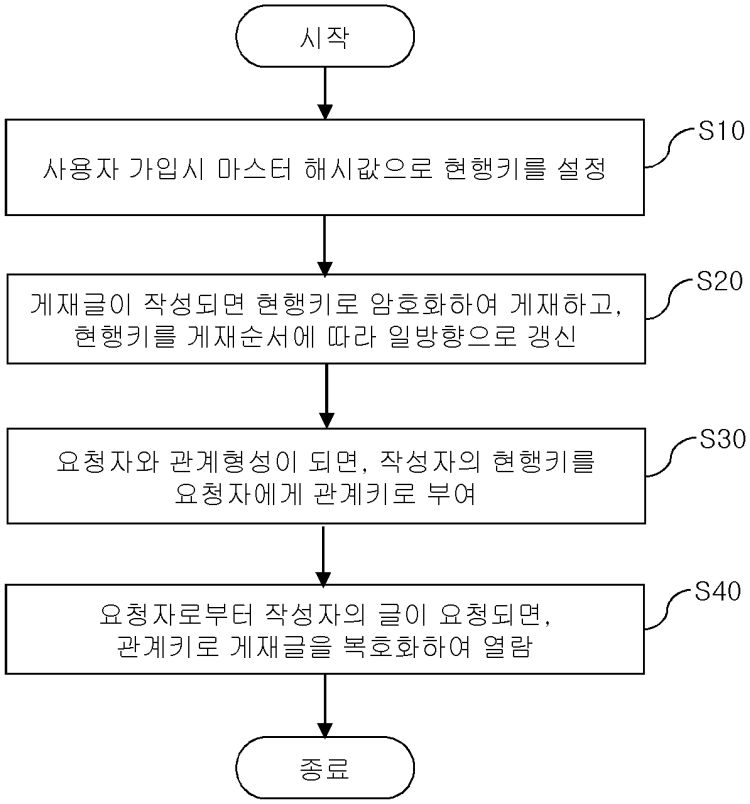
도면8

추가로 저장되는 정보	데이터	설명
$master\_h_{BA}$	$=cur\_h_A$	사용자B가 사용자A와 관계를 형성하였을 때의 마스터 cur_h_{BA} (마스터 관계키)
cur_h_{BA}	$=cur\_h_A$	사용자B가 사용자A의 글을 읽을 때 복호화에 사용되는 키이며, 사용자A의 글을 읽을 때마다 해시 함수를 이용하여 업데이트 됨 (관계키)
$master\_count_{BA}$	$=count_A$	사용자B가 사용자A와 관계를 형성하였을 때의 마스터 $count_{BA}$
$count_{BA}$	$=count_A$	사용자A가 작성한 글 중 사용자B가 읽은 마지막 글의 일련번호

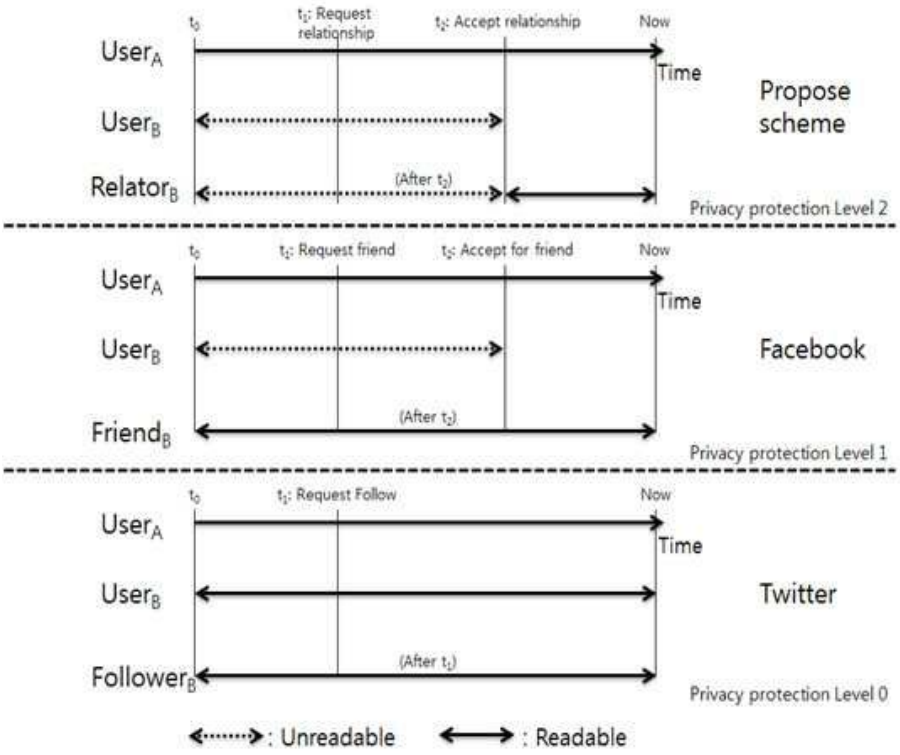
도면9



도면10


시작
사용자 가입시 마스터 해시값으로 현행키를 설정
S10
게재글이 작성되면 현행키로 암호화하여 게재하고, 현행키를 게재순서에 따라 일방향으로 갱신
S20
요청자와 관계형성이 되면, 작성자의 현행키를 요청자에게 관계키로 부여
S30
요청자로부터 작성자의 글이 요청되면, 관계키로 게재글을 복호화하여 열람
S40
종료

도면11


User$_A$
User$_B$
Relator$_B$
Time
Propose scheme
Privacy protection Level 2
User$_A$
User$_B$
Friend$_B$
Time
Facebook
Privacy protection Level 1
User$_A$
User$_B$
Follower$_B$
Time
Twitter
Privacy protection Level 0
: Unreadable
: Readable

도면12

	SHA-256	AES-CBC(128)
가입	2	0
글 작성	1	1
글 읽기	1	1
합계	4	2

(a)

(단위: microsecond)

	해시	암/복호화	합계
가입	0.550	0	0.550
글 작성	0.275	1.794	2.069
글 읽기	0.275	1.794	2.069
합계	1.100	3.588	4.688

(b)