+ 김형중 · 고려대학교 정보경영공학부 교수

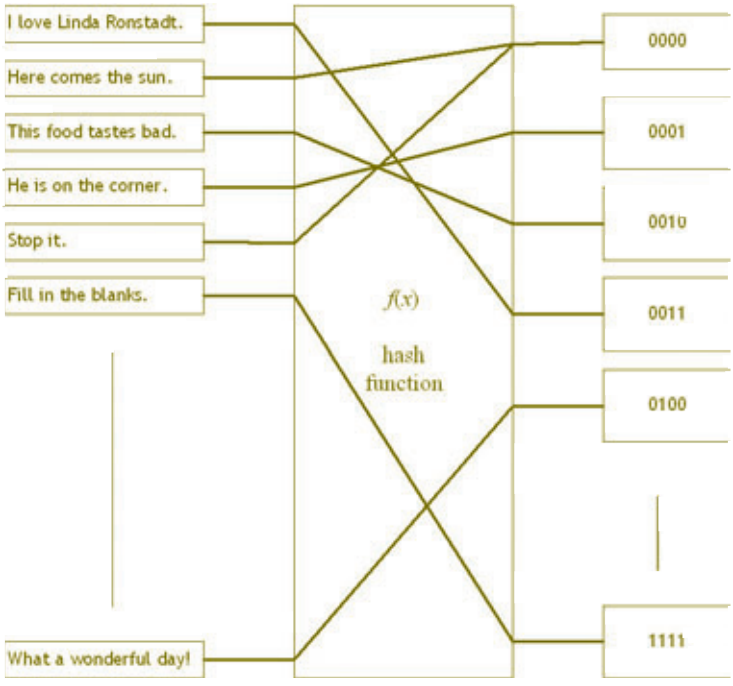
TECH&TREND

콘텐츠보호 기술의 현황과 전망 : 해쉬 기술

무결성(integrity)

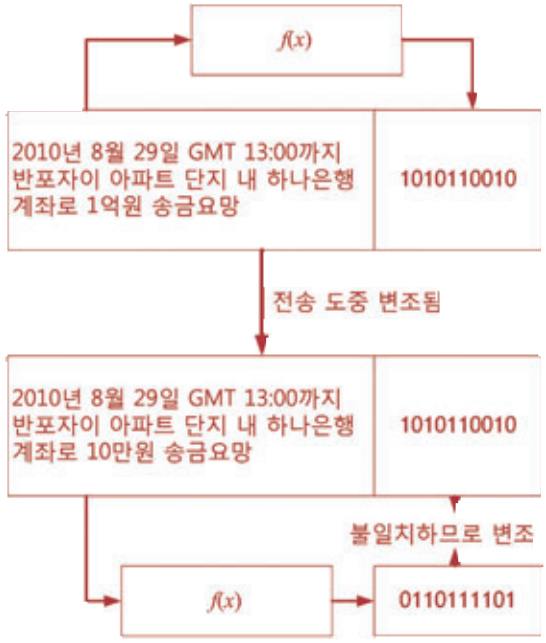
컴퓨터공학과에 다니는 인수는 집에 급히 문자 메일을 보냈다. 보안공학 원서를 한 권 사야 하는데 10만원이 필요하다고. 그런데, 전송 도중 묘하게 오류가 발생했다. 그럼에도 불구하고 TCP 프로토콜은 에러를 발견하지 못했고 그래서 재전송 요구도 없었다. 문자를 받은 아버지는 아들이 90만원을 요청하다니 놀라기도 했지만 한편으로는 90만원짜리 책을 사서 본다는데 기쁘기도 했다. 평생 자발적으로 공부하겠다고 한 적이 없는 아들이 대견스러워 즉시 송금했다. 이 시나리오에서는 악의가 개입되지 않았기 때문에 큰 문제가 없다. 착한 아들 인수라면 80만원을 돌려줄 테니까.

한편, 외교관인 인숙은 중대한 일이 발생해 본국에 1억원 송금을 급히 요청했다. 그런데, 중간에서 적국의 요원이 메일을 가로채 1억원을 10만원으로 고쳐 보낼 수 있다. 본국에서는 10만원을 급히 요청하다니 뭔가 이상하기는 하지만 그럴 수도 있겠다 싶어 의심하지 않고 송금했다고 치자. 현지에서 발을 동동 구르던 인숙은 10만원을 받고 아연 실색하지 않을 수 없다.



[그림 1] 해쉬의 입력과 출력을 연결해주는 해쉬 함수의 개념도

그렇다면 본국의 외무부에서 수신한 메일이 변조되었는지 알 길은 없을까? 변조된 메일은 무결성이 훼손되었다고 말한다. 데이터 무결성은 데이터가 허가되지 않은 주체에 의해 삭제, 수정, 추가되었는지 확인 할 때 사용한다. 즉, 그런 변조행위를 감시 또는 방지하기 위해 사용한다. 무결성을 검증 할 때 널리 사용하는 것이 암호학적 해쉬(hash)이다.



[그림 2] 해쉬를 이용해 무결성을 검증하는 예

해쉬 함수

해쉬는 다른 말로 표현하면 요약이라고 할 수 있다. 입력의 길이에 무관하게 일정한 길이의 요약을 만들 수 있다. 예를 들어, 1M 바이트의 책이든 1K 바이트의 메일이든 512비트로 요약 할 수 있다. 그것이 길다면 256비트 또는 128비트로 줄일 수 있다. 해쉬는 요약한 것이므로 입력의 집합이 출력의 집합보다 사이즈가 크고 원소의 수가 많으리라는 것은 자명하다. 그러므로 여러 입력이 한 출력으로 몰릴 수 있다. [그림 1]에서 보면 서로 다른 두 입력이 모두 출력 0000으로 몰린 것을 볼 수 있다. 이와 같은 현상을 해쉬 충돌이라고 부른다. [그림 1]에서는 해쉬를 4비트 길이로 요약한 것을 가정했다.

출력 집합의 크기가 작기 때문에 해쉬 충돌은 피하기 어렵다. 그러나, 해쉬 충돌이 고루 발생한다면 바람직하다. 한 곳에 몰리는 현상은 암호학적으로 위험하다. 극단적으로 잘못 설계된 해쉬가 있어 모든 입력에 대한 해쉬 값이 한 곳에 다 쏠렸다고 가정해보자. 이럴 경우 해쉬는 있으나마나 하게 된다. 공격하는 사람도 그 해쉬의 취약점을 알 경우 안심하고 원문을 변조 할 수 있다.

해쉬 값을 생성 할 때 사용하는 함수는 특별한 성질을 지녀야 한다. 입력 x가 주어졌을 때 출력 f(x)를 계산하는 것은 쉽고 빨라야 한다. 그러나, f(x)로부터 x를 알아내는 것은 매우 어렵고 힘들어야 한다. 예를 들어, 입력으로 ""I love Linda Ronstadt.""가 주어졌을 때 이것으로부터 출력 0011을 만드는 과정은 단순해야 한다. 그런데, 0011로부터 ""I love Linda Ronstadt.""를 찾아내는 것은 어려워야 한다. 물론, 이 과정은 비가역적 과정이므로 0011로부터 원래의 문장을 찾는 것은 쉽지 않다. ""I love Linda Ronstadt.""로부터 0011이 만들어질 수도 있고 ""I love Linda Ronstadt, but I didn't.""로부터 0011이 만들어질 수도 있다. 그게 전부가 아니다. 무수히 많다.

다시 인숙은 ""2010년 8월 29일 GMT 13:00까지 반포자이 아파트 단지 내 하나은행 계좌로 1억원 송금요망""이라는 메시지를 보내면서 이 메시지의 해쉬 값을 계산해서 그 값도 함께 보낸다. 해쉬를 생성 할 때는 암호학적 키를 사용한다. 이제 인숙의 메시지를 도중에 적국의 요원이 가로채 1억원을 10만원으로 고치기는 했는데 해쉬 값은 고치지 못한 채로 보내면 본국의 외무부에서는 인숙의 메지가 변조되었음을 바로 알 수 있다. 인숙의 메시지를 입력으로 삼아 해쉬를 생성해서 전송된 해쉬 값과 비교해보니 둘이 서로 일치하지 않기 때문에 전송된 메시지는 변조된 것이 분명하다([그림 2] 참조).

그렇다면 공격자는 정확한 해쉬를 왜 생성 할 수 없을까? 인숙과 본국의 외무부가 사용하는 함수f(x)를 모르기 때문이다. 메시지인 x는 알지만 함수를 모르므로 원본은 변조 할 수 있어도 해쉬 값은 고칠 수 없다. 혹여 f(x)를 안다고 해도 사용한 키 k를 모르면 해쉬를 생성 할 수 없다. 그러므로 해쉬는 그대로 두고 메시지만 변조해봐야 소용이 없다.

해쉬 함수나 사용한 키를 모르고 메시지만 변조하면 위에서 본 바와 같이 쓸모가 없게 된다. 그렇다면 그럴 법한 해쉬 값으로 얼렁뚱땅 바꿔치기 하는 수가 있다. 그러나, 이것은 무모한 시도에 불과하다. 512비트 해쉬를 쓸 경우 적당히 찍은 해쉬 값이 요행히 일치 할 확률은 2512분의 1. 즉, 0에 가깝다. 이것은 2010년 월드컵 결과를 모두 맞췄다는 독일의 문어가 512번의 경기결과를 다 맞출 확률과 같은 것이다.

멀티미디어 해쉬

해쉬의 사용분야가 무결성 검증으로 국한되는 것은 아니다. 특정한 입력의 존재여부를 확인 할 때도 쓸 수 있다. 이런 목적으로 사용하는 해쉬는 블룸 필터(Bloom filter)라고 부른다. 해쉬를 써서 음란물을 차단 할 수 있다. 저작권 보호에도 쓸 수 있다. 해쉬를 이용해 검색도 할 수 있다. 해쉬는 그 용도에 맞게 생성하는 방법이 달라야 한다. 무결성 검증에 사용 할 해쉬와 검색에 사용 할 해쉬는 만드는 방법이 다르고 사용하는 방법도 다르다.

텍스트는 압축될수록 가치가 높다. ""왜 사냐건 웃지요""나 ""내가 너의 이름을 불러주었을 때 너는 나에게로 와 꽃이 되었다""는 시적표현을 보자. 지금 그 상태에서 조금만 더 문장이 늘어나거나 글자가 추가되면 감동이 완전히 사라질 수 있다. 더 줄여도 그렇다. 그러니 텍스트는 인접한 단어나 글자와 거의 상관관계가 없다. 시는 그렇다. 장편소설이라면 상황은 약간 달라진다. 10권으로 구성된 대하 장편소설을 1권으로도 줄일 수 있고, 1쪽으로도 줄일 수 있다. 이렇게 줄일 때는 암호학적 해쉬 기법을 적용하지는 않는다. 암호학적 요약은 오히려 텍스트 검색을 어렵게 할 뿐이다. 짧은 문장 하나나 중요한 키워드 몇 개가 검색에는 더 유용하다. 햄릿에서는 ""to be or not to be""라는 구절만으로도 또는 주인공 이름 몇 개로 비교적 정확하게 원하는 검색성과를 달성 할 수 있다.


[그림 3] 오륙도 사진

그러나 영상이나 비디오는 키워드로 요약하기 어렵다. 멀티미디어에는 기본적으로 키워드라는 개념이 존재하지 않는다. 검색을 쉽게 하기 위해 태그를 만들자는 제안도 있었다. 2009년 부산관광사진전국 공모전 대상 작품인 [그림 3]의 풍경 사진은 나중에 합성으로 밝혀졌다. 오륙도 풍경사진에 따로 촬영한 갈매기 두 마리를 컴퓨터로 합성했던 것이다. 이 사진 속의 붉은 부리 갈매기는 겨울 철새로 촬영했을 때인 7월에는 머리의 색이 진한 갈색을 띤다고 한다. 바로 이 사실 때문에 사진 합성의 결정적인 증거가 되었다. 그런데, [그림 3]을 키워드 하나로 요약해야 한다고 하자. 그래서 혹자는 갈매기, 혹자는 오륙도, 혹자는 바다, 혹자는 대상, 혹자는 변조된 사진이라고 요약하려 할 것이다. 만일, ""변조된 사진""으로 태그되어 있다면 이 사진의 배경을 전혀 모르는 사람이 갈매기나 오륙도라는 키워드로 검색하면 이 사진을 찾는 것은 불가능하다. 이것이 검색의 어려움이나 그 한계를 보여준다. 물론, ""갈매기""로 태그가 설정되어 있었다면 찾을 수 있겠지만 수십 만 개의 사진 가운데 하나라서 하나씩 클릭해 직접 확인하다 보면 찾는데 시간이 좀 걸릴 것이다.

비디오 하나의 용량이 워낙 크다 보니 일반적으로 원본을 통째로 저장하거나 인터넷에서 그대로 전송하지 않는다. 대개 압축해서 저장하거나 전송한다. 그런데, 압축해도 용량이 대하소설보다 몇 배 크다는 문제가 발생한다. 그래서, 영상이나 비디오를 요약 할 필요가 생긴다. 소설이나 논문 등은 키워드로 요약을 대신 할 수 있으나 영상이나 비디오에서는 키워드라는 게 없기 때문에 다른 방법을 써야 한다. 그래서, 필요한 것이 멀티미디어 검색용 해쉬이다. 무결성 검증이 목적이라면 MD5나 SHA-1 등 표준화된 해쉬를 쓰면 된다.

멀티미디어를 입력으로 삼아 검색용 해쉬를 만드는 방법에는 어떤 것이 있을까? 영상의 경우 그 영상을 하나의 행렬로 보고 특이치(singular value)를 구하는 방법, 그 행렬을 음이 아닌 행렬들로 분해하는 방법(non-negative matrix factorization method), 히스토그램을 이용하는 방법 등 다양하다. 일반 행렬을 대칭행렬로 만들면 그 대칭행렬의 고유치(eigenvalue) 가운데 가장 큰 것을 특이치라 부른다. 복잡한 과정을 거쳐 만든 특이치는 그 영상의 특징을 잘 표현하는 성질이 있다. 영상이 비슷하다는 것은 행렬로 표현했을 때 행렬의 값도 비슷 할 것이므로 특이치도 비슷 할 것이라는 가정에서 출발한 것이다. 그러나, 이런 방법은 계산시간이 많이 걸리는 단점이 있고 양성오류도 크다.

검색의 어려움

특이치를 이용해 검색해보면 재미있는 현상을 목격 할 수 있다. 전혀 비슷하지 않은 영상도 비슷하다고 판정해준다. 해쉬 충돌이 발생 할 수 있음은 이미 앞에서 지적한 바와 같다. 즉, 비슷하지 않은 것도 비슷하게 분류 될 가능성이 있다는 것이다. 당연히 비슷하지 않은 영상도 비슷하다고 하면 검색에서는 놓치지 않게 된다. 극단적인 예를 들어보자. 모든 영상의 해쉬 값이 모두 한 값으로 집중된다면 그 해쉬 값으로 검색하고자 할 경우 모든 영상이 검색대상이 된다. 그러므로 찾고자 하는 영상은 당연히 후보 그룹에 포함되어 있으니 놓칠 리 없다. 인내심을 갖고 하나씩 모든 영상을 눈으로 직접 확인하다 보면 끝내 찾을 수 있다. 그러나, 그런 해쉬는 있으나마나 하다. 검색 할 범위를 최대한 좁혀 주는 해쉬야말로 정말 유용한 해쉬인 것이다. 점쟁이에게 누가 범인이냐고 물었더니 한국에 있는 국민 가운데 누군가가 범인이라고 말하는 것과 진배없다.

바람직한 해쉬는 비슷한 것은 비슷하다고 해주고 틀린 것은 틀렸다고 해주는 것이다. 그런데, 음란물을 검색하는 필터를 쓸 경우 실제로 음란물이 아닌 것도 음란물로 판단하는 경우가 있다. 그런데, 반드시 전달되어야 할 중요한 메일인데 스팸으로 처리해버릴 경우 심각한 문제가 발생 할 수 있다. 그래서, 이런 문제를 해결하기 위해 이번에는 음란물 판단 기준을 완화 할 경우 음란물이 걸러지지 않고 들어와 부모를 당혹하게 만들 수 있다. 필터링을 신이 하는 것이 아닌 만큼 늘 이런 오류가 발생한다. 음란물이 아닌데 음란물이라고 판정하는 것. 즉, 정상인데 사고로 간주하는 것을 양성오류(false positive)라고 한다. 반대로 음란물인데 음란물이 아니라고 하는 것을 음성오류(false negative)라고 말한다.

검색용 해쉬가 무결성 검증용 해쉬와 다른 점은 변형을 허용하는 것과 허용하지 않는 것의 차이에 있다. 검증용 해쉬는 원본의 점 하나라도 고쳤으면 잡아내려는 것이므로 당연히 어떠한 변형도 허용하지 않는다. 원본을 줄이는 것도 안 되고 마침표 하나 삭제하거나 추가하는 것도 허용되지 않는다. 그것은 명백한 변형이다. 그러나, 검색용은 다르다. 원본을 줄였거나 늘였거나 마침표를 넣거나 빼거나 홍길동전은 여전히 홍길동전이다.

구글에서 ""to be or no to be""로 찾아도 ""to be or not to be""와 유사한 검색 결과를 보여준다. 구글에서 ""Britney Spears""를 검색 할 때 무려 48가지나 되는 철자의 오류를 범한다고 한다(I. H. Witten, M. Gori, and T. Numerico, Web Dragons, Morgan Kaufmann Publishers, 2007). 그래도 검색이 되게 해준다. 이것은 좋게 말하면 검색엔진이 똑똑하다는 뜻이고 나쁘게 말하면 지나치게 많은 양성오류를 허용한다는 뜻이다.
대개 검색엔진은 음성오류를 줄이기 위해 양성오류에는 관대하다. 양성오류에 너무 관대하다 보면 검색엔진을 쓸 이유가 없을 정도로 쓰레기들을 찾아줄 수 있다. 그러나, 구글은 잘못 입력한 Britney Spears의 틀린 철자에 대한 빈도. 즉, 통계를 지니고 있다. 그래서, 그 경우 비교적 정확하게 찾아준다. 왜냐하면, 그 가수보다 더 유명한 사람에 관한 정보는 거의 없기 때문이다. 필자의 이름인 ""김형중""을 입력하면 거의 가수 김형중에 대한 결과만 보여준다. 아마도 검색순위가 한참 뒤지는 곳 어딘가에 필자에 대한 정보도 있겠지만 굳이 거기까지 찾아보지 않았다. 위안이 되는 것은 영어로 ""HyoungJoong Kim""을 치면 가수 김형중에 대한 정보는 거의 나오지 않고 필자에 관한 정보가 먼저 나온다. 필자의 딸이 어느 날 어찌 된 일이냐고 물어 보길래 알아낸 사실이다.
멀티미디어도 압축을 허용한다. 요즘은 MPEG 압축이 보편화되어 있다. 압축은 그래도 원본의 중요한 성질을 그대로 잘 보존하고 있다. 그런데, 멀티미디어의 경우 편집, 가공 등이 흔히 이루어진다. 비디오의 경우 압축한 후 자막을 넣기도 한다. 사진의 경우 패러디하기 위해 말풍선을 집어넣기도 한다. 광고를 만들기 위해 사진의 일부만 잘라다가 다른 영상에 붙여 넣기도 한다. 멀티미디어도 ""Britney Spears""의 예와 같이 양성오류에 관대해질 수밖에 없다.

DNA

DNA는 생명체를 식별하는 중요한 마커이다. 사람마다 DNA가 달라 이것을 지문처럼 쓸 수 있다. 멀티미디어 식별에도 식별자가 필요하다. 해쉬나 워터마크를 식별자로 사용 할 수 있다. 이런 식별자를 요즘 DNA라고 부른다. 해쉬를 사용 할 경우 별도의 정보를 멀티미디어 파일에 숨길 필요가 없다. 아무 때라도 멀티미디어의 해쉬를 계산하기만 하면 된다. 계산된 해쉬 값을 미리 데이터베이스에 저장해둔 원본 해쉬 값과 비교하기만 하면 된다. 완전히 일치하는 해쉬 값이 있다면 같은 멀티미디어일 가능성이 있다. 이 경우에는 같은 것은 같다고 말하는 것이므로 문제가 없다. 물론, 양성오류로 틀린 것인데도 같다고 할 수 있다. 그건 문제가 된다.
그렇다. 실제로는 둘이 다를 수 있다. 해쉬를 쓸 때는 충돌이 일어날 수 있기 때문이다. 저작권이 있든 없든 웹하드에 영상을 올리려는 사람들이 많다. 특히, 월드컵 경기에서 승리하는 날에는 박지성 동영상이나 이영표 동영상을 만들어 수 없이 올린다. 김연아가 우승하는 날도 마찬가지다. 동일한 동영상이 무수히 올라오게 되면 웹 하드도 감당하기 어렵게 된다. 그래서, 해쉬를 이용해 같은 동영상 업로드를 봉쇄하기 시작한다. 그런데, 박지성 동영상과 같은 것만 봉쇄하면 다행인데 엉뚱한 동영상도 해쉬 값이 같다는 이유로 차단시킬 수 있다.
해쉬를 쓸 때는 원본들에 대한 해쉬 값들을 미리 계산해서 데이터베이스에 저장해 두어야 한다. 그렇지만 해쉬 값을 영상에 숨기지는 않는다. 그러므로 흘러간 옛 영화든 최신 비디오든 언제고 차단 할 수 있다. 그러나, 워터마크는 다르다. 미리 워터마크를 숨겨두어야 나중에 검색하든 차단하든 할 수 있다. 따라서, 흘러간 옛 영화에는 워터마크를 넣지 않았기 때문에 지금이라도 워터마크를 숨겨야 한다. 이 경우 원본이 약간 훼손되는 것은 감수해야 한다. 해쉬의 경우 원본에 숨기는 것이 없기 때문에 원본 훼손도 없다.

원본의 해쉬 값은 데이터베이스에 저장한다. 그러나, 변형된 영상의 해쉬는 미리 저장하지 않는다. 미리 저장 할 수 없다 기보다 미리 계산 할 수 없다. 어떻게 변형할지 모르기 때문에 미리 해쉬를 만들어 저장해봐야 소용이 없다. 그 대신 원본의 해쉬와 비슷하다면 비슷한 영상이겠거니 하고 쓸 뿐이다. 미리 변형 할 수 있는 길은 무수하다. 그러므로 예상 가능한 모든 공격에 대비한 해쉬를 미리 만든다는 것은 전혀 의미가 없다.

워터마킹과 같이 공격의 일환으로 해쉬에도 다양한 공격이 가해진다. 그리고, 해쉬를 만드는 방법에 따라 공격방법도 다 다르다. 동물이나 식물의 세계에 천적이 존재하듯 해쉬도 마찬가지이다. 예를 들어, 히스토그램을 이용해 해쉬를 만들 경우 이것의 가장 무서운 천적은 히스토그램 평활(histogram equalization) 공격이다. 원래의 히스토그램 모양을 이용해 해쉬를 생성했는데 그 히스토그램을 평평하게 만들어버리기 때문에 이 공격 후 만드는 해쉬는 쓸모가 없게 된다. 어느 방법이나 다 아킬레스건이 있게 마련이다. 게다가 비슷하다는 것의 정의도 되어있지 않다. 비슷한 정도에 대한 정의도 없을 뿐더러 크기가 비슷하다는 것인지 색이 비슷하다는 것인지 도대체 무엇이 비슷하다는 것인지에 대한 정의도 없이 비슷한 것을 찾는 방법을 만들려 하니 그게 문제인 것이다.

응용분야

해쉬는 불법동영상을 찾아내거나 웹 하드 등에 불법으로 업로드하는 것을 막는데 주로 사용한다. 한국에서는 웹 하드가 불법 동영상 거래의 온상으로 여겨지고 있다. 그래서, 웹 하드에 동영상을 올릴 때 서버에서 이를 효과적으로 차단하도록 오디오 DNA라 불리는 해쉬를 사용하고 있다. 동영상 중 오디오 부분만 따로 떼어내 오디오 해쉬 값을 계산한다. 물론, 오디오 DNA만 사용하는 것은 아니다. 비디오 DNA도 있다. 음란물을 차단 할 때는 살색을 주로 이용한다. 법원에서 인종차별 요소가 있다고 해서 살색이라는 용어를 사용하지 못하게 한 바 있다. 그것은 살색이 한민족의 보편적 피부색을 의미했기 때문이다. 아무튼 백인종이든 황인종이든 흑인종이든 피부 색깔의 분포 정도로 음란물을 차단하려 한 시도는 충분한 가치가 있다.

음란 동영상을 차단 할 때는 이미지 해쉬를 사용한다. 불법 동영상을 차단하기 위해 대형 포탈은 수백 명의 인력을 투입하고 있다. 그런데, 해쉬를 쓸 경우 직접 검색 할 동영상의 수를 크게 줄일 수 있다. 바람직하게 자동으로 모든 검색이 이루어지면 좋으련만 양성오류 때문에 어쩔 수 없이 인력을 투입해 직접 보고 판단하게 할 수 밖에 없다. 그러나, 해쉬가 제 역할을 하면 검색 대상이 되는 비디오의 양이 크게 줄어들어 인력도 줄일 수 있는 효과가 있다.

일부 언론에서 해쉬와 DNA가 다른 것처럼 보도하고 있는데 이는 그 본질을 모르기 때문이다. 초기에는 32비트 분량에 파일 크기, 파일 제목 등의 정보를 담았던 것을 것은 해쉬라 말하는데 엄밀히 말하면 그것은 태그에 해당된다. 여기서 말하는 DNA나 해쉬는 둘 다 지각적(perceptual) 해쉬라고 불리는 것이다. 지각적으로 듣거나 봤을 때 원본과 동일하다고 판단되는 것을 해쉬로 만들었다는 뜻이다. 해쉬는 그 종류가 너무 많아서 일일이 구분하기도 어려운데 그 가운데 암호학적 해쉬와 지각적 해쉬의 성질과 쓰임새는 완전히 다르다. 그리고, 해쉬와 DNA는 저작권보호 분야에서는 같은 의미로 쓰인다.

구글은 순서(order)의 중요성을 깨달아 검색시장에서 성공했다. 사용자가 찾고자 하는 항목에 가장 근접했다고 보여 지는 검색결과를 가장 먼저 보여준 것이 구글의 성공비결이다. 사용자가 무엇을 찾고자 하는지를 정확히 아는 것은 불가능하다. 앞서 말한 바와 같이 무엇이 비슷한지에 대한 정의도 없듯 사용자가 무엇을 원하는지에 대한 정의도 없다. 그래서, 원하는 것을 찾아준다는 것은 신의 영역이다. 그러나, 사람들은 비슷한 생각을 하기도 하므로 이런 성질을 이용해 검색결과 순서를 정하는 것이다. 지금 우리는 페이지랭크(PageRank)라는 구글의 알고리듬을 수용하고 있을 뿐이다.

다음 세대의 검색엔진은 영상이나 비디오 일부로 유사한 것을 찾아주는 서비스를 포함해야 할 것으로 예상하고 있다. 그때 검색용 해쉬가 중요한 역할을 할 것으로 보인다. 그렇지만 저작권보호를 위해 검색용 해쉬가 음란물이나 저작권이 있는 비디오를 차단하는 일에 널리 쓰이고 있다.

이 글은 문화체육관광부 및 한국콘텐츠진흥원의 2010년도 문화콘텐츠산업기술지원사업(CTRC)의 연구결과로 수행되었다.