DOI: 10.3745/KIPSTC.2009.16-C.1.27

가중치 테이블 기반 안전한 e-비즈니스 데이터 분할 복원 방식

송 유 진[†] · 김 진 석[††]

요 약

최근의 개인정보 노출의 가장 큰 원인은 정당한 개인정보 관리자 즉, 내부자에 의한 부정 이용이다. 관리자는 사용자의 기밀문서를 몰래 복사하거나 고쳐 쓰는 것도 가능하다. 이러한 민감한 개인 및 기업 데이터의 안전한 관리 대책으로서 기밀정보의 안전한 분산 저장관리 기능이 요구되고 있다. 한편, 개인정보를 제공하는 경우, 정보 소유자가 프라이버시 데이터를 누구에게 얼마만큼 가중치(weight)을 두고 공개할지를 정해야 한다. 따라서 참여자의 중요도에 따라 개인정보 열람에 대한 권한의 가중치를 다르게 지정할 수 있는 구조가 요구된다. 본 논문에서는 개인정보를 권한의 가중치에 따라 안전하게 관리할 수 있는 새로운 데이터 관리 기법을 위한 Digit-independent 알고리즘을 새롭게 제안한다. 제안된 알고리즘을 근거로 데이터의 수집과 단순한 연산만으로 고속 연산처리가 가능하여 대량의 프라이버시 데이터에 적용할 수 있는 데이터 관리기법으로서 가중치 테이블 기반의 Digit-independent 알고리즘을 이용한 데이터 분할 복원 방식을 제안한다. 제안 방식은 유비쿼터스 환경에서 e-비즈니스 데이터의 안전한 관리 및 저장에 활용될 수 있을 것이다.

키워드 : 개인정보, 분할/복원, 가중치 테이블, Digit-independent 알고리즘

Secure Sharing and Recovering Scheme of e-Business Data Based on Weight Table

Song Youjin[†] · Kim Jinseog[††]

ABSTRACT

The leaking of personal information is mostly occurred by internal users. The confidential information such as credit card number can be disclosed or modified by system manager easily. The secure storaging and managing scheme for sensitive data of individual and enterprise is required for distributed data management. The manager owning private data is needed to have a weight which is a right to disclose a private data. For deciding a weight, it is required that system is able to designate the level of user's right.

In this paper, we propose the new algorithm named digit-independent algorithm. And we propose a new data management scheme of gathering and processing the data based on digit-independent algorithm. Our sharing and recovering scheme have the efficient computation operation for managing a large quantity of data using weight table. The proposed scheme is able to use for secure e-business data management and storage in ubiquitous computing environment.

Keywords : Personal Information, Sharing/Recovering, Weight table, Digit-independent Algorithm

1. 서 론

인터넷상의 온라인 상거래가 늘어나면서 민간사업자에 의한 개인정보의 수집이 늘어나고 있으나 개인정보보호 의식이 저조하여 저장된 개인의 정보가 악의적인 해킹에 노출되는 경우가 많다. 또한, 공공기관의 경우도 개인정보보호를 위한 보안관리가 허술하고 과도한 개인정보 열람 등으로 프라이버시를 침해하고 있는 실정이다.

최근 고객관계관리(Customer Relationship Management: CRM) 등이 기업의 경쟁력을 갖추기 위한 지식경영의 필수요소로 등장하면서 기업의 개인정보 수집 및 이용이 증가하고 있으나 민간 사업자 또는 공공기관의 개인정보보호 의식이 저조하여 저장된 개인의 정보가 악의적인 해킹에 노출되는 경우가 많다. 예를 들어 사용자가 어떤 제품을 얼마나 오래 사용하였는가와 같은 사용자의 기호에 대한 정보가 악

※ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No. R01-2008-000-20062-0(2008))
† 정 회 원 : 동국대학교 정보경영학과 교수
†† 정 회 원 : 동국대학교 정보통계학과 조교수
논문접수 : 2008년 8월 26일
수 정 일 : 1차 2008년 10월 9일, 2차 2008년 11월 1일
심사완료 : 2008년 11월 3일

의적인 정보수집자에게 노출될 경우, 마케팅에 불법적으로 악용될 우려가 있다. 한편, 유무선 인터넷을 기반으로 하는 u-헬스케어 서비스가 활성화되면서 개인의료 데이터의 수집, 저장 및 처리 등이 용이하게 되어 개인의 민감한 의료정보의 노출이 사회적인 문제로 대두될 수 있게 되었다.[1]

이와 같이 유비쿼터스 컴퓨팅 환경이 도래함에 따라 개인정보의 수집범위는 크게 증가하였다. 기업들은 비즈니스의 활성화를 위하여 이러한 대량의 개인정보를 수집 및 관리하게 될 것이다. 안전한 유비쿼터스 환경의 실현을 위해서는 프라이버시 보호에 대한 고려가 필수적이며, 이에 적합한 개인정보 관리 방법이 필요하다.

최근의 개인정보 노출의 가장 큰 원인은 정당한 개인정보 관리자 즉, 내부자에 의한 부정 이용이다. 관리자는 사용자의 기밀문서를 몰래 복사하거나 고쳐 쓰는 것도 가능하다. 사용자 키가 로그인 패스워드라면 그 사용자를 위장하는 것도 가능하게 된다. 이와 같이 접근 권한을 한곳에 집중시키면 관리는 쉽지만 이에 따르는 리스크가 커지게 된다. 이러한 민감한 개인 및 기업 데이터의 안전한 관리 대책으로서 기밀정보의 안전한 분산 저장관리 기능이 요구되고 있다.

개인정보의 안전한 관리를 위한 대책으로서 Shamir 비밀분산(Secret Sharing) 방식[2]이 있다. 기존의 비밀분산방식은 대용량의 연산이 요구됨에 따라 대용량 데이터에 대한 취급에는 적절하지 않고 연산과정에서 많은 메모리가 요구되어 ISP(Information Service Provider) 등의 사업자가 저장하고 있는 대량의 개인정보 데이터 처리에 적합하지 않다. 예를 들어 수직선상의 y절편에서 해를 찾는 방식의 경우, 지정된 두 점간의 선을 구하기 위하여 연립방정식에 대한 연산이 요구된다. 이러한 방법은 대량의 개인정보 관리에 적절하지 않고 분할 데이터에 따른 가중치를 고려할 수 없어 프라이버시 데이터의 중요도에 따라 관리할 수 없다.

적절한 권한을 가진 내부 사용자는 필요에 따라 사용자에 대한 개인정보를 합법적이 아닌 방법으로 취득할 수 있다. 이러한 경우 불법적으로 취득한 정보를 유출 및 이메일 전송을 하는 경우, 개인정보 유출에 대한 피해가 막대할 수 있다. 한편, 개인정보를 제공하는 경우, 정보 소유자가 프라이버시 데이터를 누구에게 얼마만큼 가중치(weight)을 두고 공개할지를 정해야 한다. 따라서 참여자의 중요도에 따라 개인정보 열람에 대한 권한의 가중치를 다르게 지정할 수 있는 구조가 요구된다. 이러한 요구사항은 개인정보가 기록된 분할 데이터에 대해 각각의 가중치를 별개로 두고, 특정 권한을 가진 관리자(예를 들면, 정보보호 책임자 등)은 몇 개의 분할 데이터를 더 수집하는 것만으로 복구가 가능한 가중치를 적용하는 방식으로 해결할 수 있다.

본 논문에서는 개인정보를 권한의 가중치에 따라 안전하게 관리할 수 있는 새로운 데이터 관리 기법을 위한 Digit-independent 알고리즘을 새롭게 제안한다. 제안된 알고리즘을 근거로 데이터의 수집과 단순한 연산만으로 고속 연산처리가 가능하여 대량의 프라이버시 데이터에 적용할 수 있는 데이터 관리기법으로서 가중치 테이블 기반의 Digit-independent 알고리즘을 이용한 데이터 분할 복원 방식을 제안한다.

본 논문의 결과는 프라이버시 데이터 관리 연구의 새로운 유형이 될 것이며, 기존 이론의 발전과 함께 새로운 해석을 통한 고유한 이론 체계를 창출할 수 있을 것이다. 특히 제안된 분할 복원 모델을 통해서 대용량의 고속연산이 가능한 개인정보 관리 기법을 도출하는 실용적인 결과를 얻을 수 있다. 제안 방식은 유비쿼터스 환경에서 e-비즈니스 데이터의 안전한 관리 및 저장에 활용될 수 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서 본 연구와 간접적인 관련성이 있는 Shamir 의 비밀분산방식을 검토하고 3장에서는 논문에서 제안하는 방식을 서술한다. 그리고 4장에서는 제안 방식에 대한 분석을 행하고 마지막으로 5장에서 결론을 맺는다.

2. Shamir 비밀 분산 방식

리스크 분산을 개인정보 관리에 적용했을 경우, 예를 들면, 중요한 개인정보를 3개의 분할 정보로 나누어 3사람의 관리자에게 각각 맡기는 시스템을 가정한다. 개인정보를 복원하고 싶을 때, 관리자 3인이 자신의 가지고 있는 분할 정보를 모으면 키를 복원할 수 있다. 이러한 처리는 실제 비밀분산 방식이라고 하는 k-out-of-n 분산 방식으로서 「분할시키는 수 : n」, 「복원에 필요한 수(임계치): k」인 방식이 있다.[2]

비밀분산법은 비밀 정보를 다수의 분할 데이터로 분할한 후 복원 권한을 가진 구성원이 분할 데이터를 수집하여 복원함으로써 데이터를 안전하게 관리하는 프로토콜이다. 비밀분산방식은 불특정 다수 혹은 저장소에서 비밀정보를 관리하는 이 방법은 연산속도상의 문제를 갖고 있다.

비밀분산 방식은 Blakely[3]와 Shamir[2]에 의해 제안되었다. 그 중 다항식 보간법을 이용하는 Shamir의 비밀분산 방식은 다음과 같다.

- p : $p \geq n+1$인 큰 소수(단, n은 비밀분산에 참여하는 전체 참가자의 수)
- K : 분산하고자 하는 비밀, $K \in GF(p)$
- D : 각 참가자에게 부분정보(개인정보 분할 데이터)을 전달하는 분배자
- P : 전체 참가자의 집합, $P=\{P_1, P_2, \ldots P_n\}$
- S_i : 참가자의 P_i에게 분배하는 분할정보

2.1 비밀분산 과정

(1) 정보분배자는 $GF(p)$상에서 0이 아닌 n개의 원소 $x_1, x_2, \ldots, x_n$을 랜덤하게 선택한다.

(2) 정보분배자는 $GF(p)$상에서 $a_1, a_2, \ldots, a_n$을 랜덤하게 선택하고 다음과 같이 $(t-1)$차 다항식을 생성한다.

$$f(x) = K + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$$

(3) 정보분배자는 각 참가자 $P_1, P_2, \ldots, P_n$에게 분배할 부분정보 $S_i (1 \le i \le n)$을 다음과 같이 계산하여 $P_i (1 \le i \le n)$에게 안전하게 전송한다.

$$S_i = f(x_i) (1 \le i \le n)$$

(4) $x_1, x_2, \ldots, x_n$는 공개하고, $a_1, a_2, \ldots, a_n$와 부분정보 $S_i (1 \le i \le n)$는 안전하게 보관한다.

2.2 비밀복원 과정

비밀 복원에 참여하는 참가자의 집합을 $P = \{P_1, P_2, \ldots P_t\}$라 한다.

(1) t명의 참가자들은 Lagrange의 다항식 보간법에 의해 $f(x)$상의 t개의 서로 다른 점 $(x_i, s_i) (1 \le i \le t)$를 이용하여 다음과 같이 $f(x)$를 계산한다.

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \le j \le t, j \ne i} \frac{x - x_j}{x_i - x_j}$$

(2) 복원하고자 하는 비밀 $K = f(0)$은 다음과 같이 계산될 수 있다.

$$K = \sum_{i=1}^{t} c_i y_i \text{ , (단, } C_i = \sum_{1 \le j \le t} \frac{x_j}{x_j - x_i} \text{)}$$

다항식 보간법을 이용한 Shamir의 비밀분산 방식은 $t (t \le n)$명 이상의 참가자들의 협조에 의해서만 본래의 비밀 K를 복원할 수 있지만 $t-1$명 이하의 참가자로부터는 K에 대해 아무런 정보도 얻을 수 없다. 이와 같은 비밀분산 방식을 (t, n) – threshold 방식이라 하며, 이러한 비밀분산 방식의 개념이 제안된 이후 여러 가지 특징을 갖는 다양한 비밀분산 방식들이 제안되었다.

Shamir에 의해 제안된 임계치 비밀분산 방식의 경우, 한 번 복원된 이후에는 참가자들의 부분 정보를 재사용할 수 없으며, 비밀데이터의 크기가 소수의 멱승 q로 제한된다. 또한, q의 증가에 따라 비밀을 분산하기 위한 함수의 수가 지수적으로 증가하여 대량 데이터의 취급에 한계가 있다.

3. 제안 방식

본 논문에서 새롭게 제안하는 가중치 테이블을 이용한 데이터 분할 복원 방식에서 쓰이는 용어는 다음과 같다.

W_m - 분할 데이터 인덱스 합이 m인 가중치
w-table - 전체 가중치의 집합
O(Original data) - 원본 데이터
L - 분할 데이터의 집합
k - 복구 과정에서 수집된 분할 데이터의 갯수
b - 버퍼 단위로 분할되는 저장공간의 개수
G - 그룹 ID의 집합
S_p - p의 고유번호를 가진 분할 데이터
n - 비밀데이터의 분할 개수

3.1 비밀데이터 분할 과정

비밀분산 절차는 원본 데이터에 대한 간단한 연산으로 분산이 가능하다. 먼저, 원본 데이터 및 전체 분할 개수, w-table을 결정한다. 이후, 원본 데이터를 2진수로 변환한 후 임계값을 설정하고 해당 버퍼값을 설정한다. 이후 버퍼에 대한 임계값 만큼의 원본값과 일치하는 데이터와 나머지 버퍼에 대한 원본값과 반대되는 데이터를 저장하는 것으로 분할 데이터 집합 L을 획득할 수 있다. 비밀데이터 분할 과정은 <표 1>과 같다.

〈표 1〉 비밀데이터 분할 과정

Input : O, n, W
Output : L = {S_1, .. , S_n}
Assume : L_k = O
Step 1 : 원본데이터를 2진수로 변환한다.
o ← $(O)_2$
Step 2 : 임계값을 설정하고, 버퍼값을 설정한다.
$t \leftarrow \frac{b}{2} + 1$, b ← $\sum_{n\,id=1}^{n} W_{n\,id}$
Step 3 : 임계값 t만큼 원본과 같은 데이터를 삽입하고, 그 이외에는 원본과 다른 데이터를 삽입한다.

임계값 t가 주어질 때, 아래의 조건을 충족시키도록 S_{pi}의 값을 부여한다고 하자. 단

$b = \sum_{p=1}^{n} W_p$, $t = b(1/2 + \Theta)$이고 $0 < \Theta < 1$이다.

(1) $o_i = 1; \Rightarrow \sum_{p=1}^{n} W_p S_{pi} > b(1/2 + \Theta)$,

(2) $o_i = 0; \Rightarrow \sum_{p=1}^{n} W_p (1 - S_{pi}) > b(1/2 + \Theta)$,

여기서 $i = 1, \cdots, L$이다.

Step 4 : p의 순서대로 데이터를 취득하여 전체 분할 데이터 집합 L을 얻는다.
L = {S_1, .. , S_n}

3.2 비밀데이터 복원 과정

비밀데이터 복원 과정은 분산 과정보다 훨씬 단순하게 계산된다. 수집한 분할 데이터의 집합과 w-table을 통하여 O을 안전하게 복원한다. 복원 과정에서 단순히 해당 가중치와 분할 데이터에 대한 곱셈의 전체 합과 버퍼의 개수를 비교하는 것만으로 원본 데이터를 복원할 수 있다.

〈표 2〉 비밀데이터 복원 과정

Input : 수집한 분할 데이터의 집합 (L_k) , W
Output : 원본 데이터(O)

Step 1 : 전체 버퍼의 크기를 구한다.

$b \leftarrow \sum_{nid=1}^{k} W_{nid}$

Step 2 : 가중치와 각 원소의 곱에 대한 합을 비교하는 것으로 결과를 구할 수 있다.

$$o = \begin{cases} 1 & \text{if } \frac{b}{2} \leq \sum_{nid=1}^{k} W_{nid} S_{nid} \\ 0 & otherwise \end{cases}$$

Step 3 : 해당 결과를 십진수로 변환하여 최초 값을 구한다.

$O \leftarrow (o)_{10}$

4. 분 석

본 장에서는 제안 방식에 대한 분석을 수행한다. 원본데이터를 복원할 수 있도록 하는 필요충분조건에 대해 정리하고 Digit-independent 알고리즘에 대해 분석한다.

4.1 원본 데이터 복원의 필요충분조건

원본데이터 O 를 길이 L인 이진수로 변환한 값을 $o=\{o_1,\cdots,o_L\}$이라고 하자. 그리고 n을 전체 분할 데이터의 개수, 즉, 원본 데이터는 $S_p(p=1,\cdots,n)$로 자료가 분할된다. 또한 $W=\{W_p,p=1,\cdots,n\}$는 각 분할데이터에 해당되는 가중치인 w-table이다.

원본데이터 및 i번째 분할데이터에 할당된 값을 아래와 같이 길이 L인 이진수로 표현하자.

$$o = \{o_i \in \{0,1\}: i=1,\cdots,L\},$$
$$S_p = \{S_{pi} \in \{0,1\}: i=1,\cdots,L\}, p=1,\cdots,n.$$

[Scheme 1] 임계값 t가 주어질 때, 아래의 조건을 충족시키도록 S_{pi}의 값을 부여한다고 하자. 단 $b=\sum_{p=1}^{n} W_p$, $t=b(1/2+\Theta)$이고 $\Theta>0$이다.

(1) $o_i=1; \Rightarrow \sum_{p=1}^{n} W_p S_{pi} > b(1/2+\Theta)$,

(2) $o_i=0; \Rightarrow \sum_{p=1}^{n} W_p(1-S_{pi}) > b(1/2+\Theta)$,

여기서 $i=1,\cdots,L$이다.

[정리] Scheme 1에 의하여 각 분할데이터에 값을 할당할 때, 가중치가 W_m인 m번째 분할데이터가 손실되더라도 나머지 분할데이터를 이용하여 원본데이터를 복원할 수 있도록 하는 필요충분조건은 $\Theta \geq \frac{1}{2} W_m / \sum_{p=1}^{n} W_p$이다.

증명: [Scheme 1]의 식(1)이 성립할 때, 모든 i에 대하여 $\sum_{p\neq m} W_p S_{pi} > 1/2 \sum_{p\neq m} W_p$, 과 [Scheme 1]의 식(2)가 성립할 때, 모든 i에 대하여 $\sum_{p\neq m} W_p S_{pi} > 1/2 \sum_{p\neq m} W_p$, 이 성립함을 보인다.

먼저 원본데이터의 i번째 이진값을 $o_i=1$, 가중치가 최소인 분할데이터에 해당되는 인덱스를 m이라고 하고 가중치는 $W_{\min}=W_m$이라고 하자.

[Scheme 1]에 의하여

$$\sum_{p=1}^{n} W_p S_{pi} = \sum_{p\neq m} W_p S_{pi} + W_m S_{mi} > b(1/2+\Theta)$$

이므로, 위 식의 오른쪽 부등식을 다시 쓰면 아래와 같다.

$$\sum_{p\neq m} W_p S_{pi} > b(1/2+\Theta) - W_m S_{mi}.$$

따라서, 부분손실자료의 복원조건인 $\sum_{p\neq m} W_p S_{pi} > 1/2 \sum_{p\neq m} W_p$ 를 만족하기 위해서는 모든 i에 대하여 $b(1/2+\theta) - W_m S_{mi} \geq 1/2 \sum_{p\neq m} W_p$ 여야 한다. 즉,

$$\min_i(b(1/2+\theta) - W_m S_{mi}) = b(1/2+\theta) - W_m \geq 1/2 \sum_{p\neq m} W_p.$$

위의 오른쪽 부등식을 다시 쓰면

(3) $$\Theta \geq \frac{1}{2} \frac{W_m}{\sum_{p=1}^{n} W_p}$$

이다.

위와 마찬가지로 $o_i=0$인 경우, 식 (2)를 다시 쓰면 아래와 같다.

$$\sum_{p\neq m} W_p S_{pi} + W_m S_{mi} < b(1/2-\theta) \rightarrow \sum_{p\neq m} W_p S_{pi} < \frac{1}{2} \sum_{p\neq m} W_p + \frac{1}{2} W_m - \theta \sum_{p=1}^{n} W_p - W_m S_{mi}.$$

따라서, 모든 i에 대하여 $\frac{1}{2} W_m - \Theta \sum_{p=1}^{n} W_p - W_m S_{mi} \leq 0$ 이여야 하고,

$\rightarrow \frac{1}{2} W_m - \Theta \sum_{p=1}^{n} W_p \leq 0$ 이므로 필요충분조건은 식(3)과 같다.

Remark 1: 만일 가중치의 전체합이 100이고 $\Theta=2/100$ 이라고 하자. 이 경우 $t=100\times(1/2+1/100)=100\times(0.51)$ 이다. 가중치가 2인 분할데이터가 손실되더라도 정확한 복원을 보장하기 위해서는 분할데이터에 값을 저장할 때, 전체의 51% 이상은 원본과 동일하도록 해야 한다.

4.2 Digit-independent 알고리즘

a. Group별로 가중치를 할당한다. 이때 전체 가중치의 합은 b이고, 각 그룹은 할당된 가중치만큼의 저장공간(buffer)을 배정받는다.

b) 분할 데이터가 손실되어도 복원될 수 있는 최소 가중치 ($W_{\min}$)를 정한다.

c) Θ를 구하고 이를 이용하여 최소 임계값 t를 구한다.

d) 각 digit별로 b개중 랜덤하게 선택한 t개의 buffer은 original digit과 동일하게 배정하고, 나머지는 반대로 배정한다.

4.1의 [정리]에 있는 것처럼 원본데이터를 복원할 수 있는 필요충분조건 $\Theta \geq \frac{1}{2} W_m / \sum_{p=1}^{n} W_p$ 에서 최소가중치가 $W_m = 1$인 경우 $\Theta \geq \frac{1}{2} \frac{W_m}{\sum_{p=1}^{n} W_p} = 1/2 \times 1/8 = 1/16$ 이므로, 임계값은 $t \geq b(1/2+\Theta) = 8 \times (1/2 + 1/16) = 4\frac{1}{2}$ 이다. 이는 아래 <표 3>의 분할 데이터 예에서처럼 13개 비트의 각 digit를 구성하고 있는 버퍼 8개 중 5개 이상은 원본 digit과 동일하도록 분할해야 한다는 것이다. 즉, 손실되어도 복원가능한 최소 가중치(W_m) 관점에서 보면, 버퍼 ($b - W_m$)개가 갖추어져야 그 중 절반이 넘는 digit이 원본과 동일하게 되어 원본 복원이 가능하다는 것을 의미한다. 따라서 Shamir의 표기법으로 표현하면, 본 논문의 방식은 ($b - W_m$)-out-of-b 이 된다.

위의 알고리즘을 아래와 같은 원본데이터(신용카드 비밀번호 7342)의 경우를 예로 든다.

source	1	1	1	0	0	1	0	1	0	1	1	1	0

아래와 같은 분할데이터를 만들 수 있다.

〈표 3〉 분할 데이터

G	W	S	이진 데이터												
			1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	1	1	1	1	0	0	1	0	1	0	1	1	1	0
		1	1	1	1	0	0	1	1	1	1	1	1	0	1
		1	1	0	0	1	0	1	0	0	0	0	0	1	0
2	2	2	1	1	1	0	0	0	1	1	1	1	1	1	0
		2	1	0	1	1	1	1	0	1	0	1	1	1	0
3	1	3	0	1	0	0	1	0	1	0	1	0	1	0	0
4	1	4	0	0	1	0	1	1	0	0	0	1	1	0	1
5	1	5	0	1	1	1	0	0	0	1	0	1	0	1	0
1의개수			5	5	6	3	3	5	3	5	3	6	6	5	2
0의개수			3	3	2	5	5	3	5	3	5	2	2	3	6
source			1	1	1	0	0	1	0	1	0	1	1	1	0

분할된 데이터로부터 특정 그룹ID G를 갖는 데이터가 분실될 경우의 복원 과정을 아래의 <표 4>로 나타내었다.

〈표 4〉 G=3이 분실될 경우의 복원절차

G	W	S	이진 데이터												
			1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	1	1	0	1	0	0	0	0	0	0	1	1	1	1
		1	1	1	1	0	0	1	1	1	1	0	1	1	1
		1	1	0	1	1	0	1	0	1	0	1	1	0	0
2	2	2	0	1	0	1	1	0	0	0	1	0	0	1	0
		2	0	0	0	0	1	0	0	1	0	1	1	0	0
3	1	3	0	1	1	0	0	1	1	0	0	0	0	0	1
4	1	4	1	1	0	0	1	1	0	1	0	1	0	1	0
5	1	5	1	1	1	1	0	1	1	1	1	1	1	1	0
복원절차		1의개수	5	4	4	3	3	4	2	5	3	5	5	5	2
		0의개수	2	3	3	4	4	3	5	2	4	2	2	2	5
		복원	1	1	1	0	0	1	0	1	0	1	1	1	0

<표 4>에서 G=3이 분실된 경우에 대한 복원 절차를 예로 든 것이다. 본 알고리즘은 그룹의 가중치가 1이상이라 하더라도 보유하고 있는 분할데이터의 하나만 분실된 경우에도 나머지 분할 데이터에서의 1의 개수가 임계값(t) 5이상이 되므로 복원이 가능하게 된다. 즉, 가중치 1인 어떤 분할데이터가 분실되던지 혹은 가중치가 2 이상인 분할 데이터에서 분실된 데이터의 가중치가 1이면 나머지 7개의 분할데이터를 이용하여 복원이 가능하다. <표 5>는 G=1에서 두 번째 값이 분실될 경우에 대한 복원 예이다.

〈표 5〉 G=1의 일부가 분실될 경우의 복원절차

B	W	S	이진 데이터												
			1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	1	1	0	1	0	0	0	0	0	0	1	1	1	1
		1	1	1	1	0	0	1	1	1	1	0	1	1	1
		1	1	0	1	1	0	1	0	1	0	1	1	0	0
2	2	2	0	1	0	1	1	0	0	0	1	0	0	1	0
		2	0	0	0	0	1	0	0	1	0	1	1	0	0
3	1	3	0	1	1	0	0	1	1	0	0	0	0	0	1
4	1	4	1	1	0	0	1	1	0	1	0	1	0	1	0
5	1	5	1	1	1	1	0	1	1	1	1	1	1	1	0
복원절차		1의개수	4	4	4	3	3	4	2	4	2	5	4	4	2
		0의개수	3	3	3	4	4	3	5	3	5	2	3	3	5
		복원	1	1	1	0	0	1	0	1	0	1	1	1	0

다양한 최소 가중치의 경우($W_{\min}$ =1,2,3)에 대한 시뮬레이션 결과는 부록에 제시하였으며, 알고리즘은 R code를 이용하여 구현하였다.

4.3 보안성 및 성능

본 논문에서는 보안성에 대해 분할데이터가 공격자에게 노출될 경우, 원본 복원 가능성을 검토한다. 제안된 방법에 의하면 노출된 버퍼의 특정위치의 digit들 중에서 원본 digit과 일치하는 digit의 개수가 1/2보다 큰 경우 원본으로 복원된다. 이에 대한 확률은 초기하분포(hypergeometric distribution)를 이용하여 표현 가능하다.
즉,

$$\Pr(\text{e개의 digit이 노출될 때 복원가능}) = \sum_{x>e/2}^{e} \frac{\binom{t}{x}\binom{b-W_{\min}}{e-x}}{\binom{b}{e}}.$$

위 식에 의한 복원가능성을 buffer 수(b)=8인 경우에 대하여 아래의 표로 정리한다.

위의 <표 6>을 보면, Case (II)에서 노출버퍼의 개수가 5인 경우를 제외하고는 완벽한 복원이 매우 어렵다. 특히, digit의 개수가 큰 코드단위에서의 복원은 사실상 불가능하다고 할 수 있다. 반면, 노출버퍼의 개수가 5인 경우는 완벽한 원본 복원이 가능하다. 이는 분할 알고리즘내에서 정수계산 때문에 발생하는 것으로 임계치(t=6) 보다 1개가 적은 경우에 발생할 수 있다. 따라서 보안성을 기하기 위해 (임계치-1) 이상의 노출이 되지 않도록 유의해야 한다.

한편, 성능 측면에서 본 논문과 Shamir 방식의 계산 속도를 비교하면 다음과 같다.

Shamir 방법은 polynomial interpolation을 이용하므로 분할데이터에 이용할 buffer의 수가 b인 경우 $O(b\log^2 b)$ 만큼의 계산을 요한다. 본 방식의 알고리즘은 digit별로 난수를 발생시키므로 원본 데이터의 전체 digit의 수를 L이라고 할 경우, $O(L)+O(b)$의 계산을 필요로 한다. 따라서 제안된 알고리즘은 buffer의 수(b)가 클 경우 계산속도가 빠르다고 할 수 있다. 물론 원본 데이터의 size(L)에 대하여도 계산속도는 선형적으로 증가하고 있음을 알 수 있다.

5. 결 론

최근의 개인정보 노출의 가장 큰 원인은 정당한 개인정보 관리자 즉, 내부자에 의한 부정 이용이다. 관리자는 사용자의 기밀문서를 몰래 복사하거나 고쳐 쓰는 것도 가능하다. 이러한 민감한 개인 및 기업 데이터의 안전한 관리 대책으로서 기밀정보의 안전한 분산 저장관리 기능이 요구되고 있다.

한편, 개인정보를 제공하는 경우, 정보 소유자가 프라이버시 데이터를 누구에게 얼마만큼 가중치(weight)을 두고 공개할지를 정해야 한다. 따라서 참여자의 중요도에 따라 개인정보 열람에 대한 권한의 가중치를 다르게 지정할 수 있는 구조가 요구된다.

본 논문에서는 개인정보를 권한의 가중치에 따라 안전하게 관리할 수 있는 새로운 데이터 관리 기법을 위한 Digit-independent 알고리즘을 새롭게 제안한다. 제안된 알고리즘을 근거로 데이터의 수집과 단순한 연산만으로 고속 연산처리가 가능하여 대량의 프라이버시 데이터에 적용할 수 있는 데이터 관리기법으로서 가중치 테이블 기반의 Digit-independent 알고리즘을 이용한 데이터 분할 복원 방식을 제안한다. 제안 방식은 유비쿼터스 환경에서 e-비즈니스 데이터의 안전한 관리 및 저장에 활용될 수 있을 것이다.

향후 과제로서는 가중치가 가변일 경우, 즉 비밀데이터 관리자의 상황(Context)을 고려한 기밀 데이터 분할 복원 방식에 대한 검토이다. 또한 프라이버시 공개율을 근거로 보안 수준에 대한 정식화 연구가 필요할 것으로 생각한다. 프라이버시 공개율을 지정하는 방법은 정보 요청자의 등급

〈표 6〉 원본 복원가능성, buffer 수(b)=8인 경우

	노출 buffer의 개수	원본과 일치해야 하는 최소 buffer의 수	원본노출확률	
			1 digit단위	코드단위 (32digits)
Case (I): $W_{\min}$=1 임계치(t)=5	1	1	0.6250	2.9387e-07
	2	2	0.3571	4.9084e-15
	3	2	0.7143	2.1081e-05
	4	3	0.5	2.3183e-10
	5	3	0.8214	0.0018
	6	4	0.6429	7.2387e-07
Case (II): $W_{\min}$=2 임계치(t)=6	1	1	0.7500	0.0001
	2	2	0.5357	2.1177e-09
	3	2	0.8929	0.0266
	4	3	0.7857	4.4511e-04
	5	3	1	1
Case (II): $W_{\min}$=3 임계치(t)=6	1	1	0.7500	0.0001
	2	2	0.5357	2.1177e-09
	3	2	0.8929	0.0266
	4	3	0.7857	4.4511e-04

이나 직무에 따라 각각 다른 가중치의 분할 데이터를 소유할 수 있도록 하는 것이며 w-Table을 통하여 사용자 프라이버시 공개율을 지정할 수 있는 구조를 명확히 하고자 한다.

참 고 문 헌

[1] 송유진, 박광용, "개인정보보호를 위한 USB 기반 분할 관리시스템 설계 및 구현", e-비즈니스연구 제9권 제2호, pp.203-221, 2008.

[2] A. Shamir, "How to share a secret", Communications of the ACM Vol.22(11), pp.612-613, 1979.

[3] G.R. Blakely, "Safeguarding cryptographic key", Proceedings of AFIPS National Computer Conference, pp.313-317, 1979.

[4] J. He, et al., "Multisecret-sharing scheme based on one-way Function", Electronic letters Vol.31(2), pp.93-95, 1995.

[5] A Beimel, "Monotone Circuits for Weighted Threshold Functions", Proceedings. Twentieth Annual IEEE, pp.67-75, 2005.

[6] H.M. Sun, "On-line multiple secret sharing based on a one-way function", Computer Communications Vol. 22(8), pp.745-748, 1999.

부 록

원본데이터의 이진 코드가 1110010101110이고 총 그룹수가 5, 버퍼의 크기가 8인 경우에 대하여 최소 가중치(=1,2,3)별 분실시나리오에 따라 원본 복원과정을 제시한다.

CASE 1: $W_{\min} = 1$ ($\theta = 0.0625$, $t = 5$)

Digit independent 알고리즘에 의한 원본데이터의 분할

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	0	1	0	0	0	0	0	0	1	1	1	1
2			1	1	1	0	0	1	1	1	1	0	1	1	1
3			1	0	1	1	0	1	0	1	0	1	1	0	0
4	2	2	0	1	0	1	1	0	0	0	1	0	0	1	0
5			0	0	0	0	1	0	0	1	0	1	1	0	0
6	3	1	0	1	1	0	0	1	1	0	0	0	0	0	1
7	4	1	1	1	0	0	1	1	0	1	0	1	0	1	0
8	5	1	1	1	1	1	0	1	1	1	1	1	1	1	0

[1-1] Group 4 's data(weight=1) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	0	1	0	0	0	0	0	0	1	1	1	1
2			1	1	1	0	0	1	1	1	1	0	1	1	1
3			1	0	1	1	0	1	0	1	0	1	1	0	0
4	2	2	0	1	0	1	1	0	0	0	1	0	0	1	0
5			0	0	0	0	1	0	0	1	0	1	1	0	0
6	3	1	0	1	1	0	0	1	1	0	0	0	0	0	1
8	5	1	1	1	1	1	0	1	1	1	1	1	1	1	0
복원			1	1	1	0	0	1	0	1	0	1	1	1	0
원본			1	1	1	0	0	1	0	1	0	1	1	1	0

[1-2] Group 5 's data(weight=1) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	0	1	0	0	0	0	0	0	1	1	1	1
2			1	1	1	0	0	1	1	1	1	0	1	1	1
3			1	0	1	1	0	1	0	1	0	1	1	0	0
4	2	2	0	1	0	1	1	0	0	0	1	0	0	1	0
5			0	0	0	0	1	0	0	1	0	1	1	0	0
6	3	1	0	1	1	0	0	1	1	0	0	0	0	0	1
7	4	1	1	1	0	0	1	1	0	1	0	1	0	1	0
복원			1	1	1	0	0	1	0	1	0	1	1	1	0
원본			1	1	1	0	0	1	0	1	0	1	1	1	0

CASE 2: $W_{\min} = 2$ ($\theta = 0.125$, $t = 6$)

Digit independent 알고리즘에 의한 원본데이터의 분할

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	0	1	0	0	0	0	0	0	1	1	1	1
2			1	1	1	0	0	1	1	1	1	0	1	1	1
3			1	0	1	1	0	1	0	1	0	1	1	0	0
4	2	2	0	1	0	1	1	0	0	0	1	0	0	1	0
5			0	0	0	0	1	0	0	1	0	1	1	0	0
6	3	1	0	1	1	0	0	1	1	0	0	0	0	0	1
7	4	1	1	1	0	0	1	1	0	1	0	1	0	1	0
8	5	1	1	1	1	1	0	1	1	1	1	1	1	1	0

[2-1] Group 2 's data(weight=2) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	1	1	0	0	1	0	1	1	1	0	0	0
2			1	1	1	0	0	1	1	0	0	1	0	1	1
3			1	1	1	0	0	1	0	1	0	1	1	1	0
6	3	1	0	0	1	0	1	1	0	1	1	0	1	0	0
7	4	1	1	0	0	1	0	0	0	1	0	0	1	1	1
8	5	1	0	1	1	0	0	1	1	1	0	1	1	1	0
복원			**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**
원본			**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**

[2-2] Group 3 's data(weight=1) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	1	1	0	0	1	0	1	1	1	0	0	0
2			1	1	1	0	0	1	1	0	0	1	0	1	1
3			1	1	1	0	0	1	0	1	0	1	1	1	0
4	2	2	1	1	0	0	0	1	0	1	0	1	1	1	0
5			1	1	1	1	1	0	0	0	0	1	1	1	0
7	4	1	1	0	0	1	0	0	0	1	0	0	1	1	1
8	5	1	0	1	1	0	0	1	1	1	0	1	1	1	0
복원			**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**
원본			**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**

[2-3] Group 3 & 4 's data(weight=2) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	1	1	1	0	0	1	0	1	1	1	0	0	0
2			1	1	1	0	0	1	1	0	0	1	0	1	1
3			1	1	1	0	0	1	0	1	0	1	1	1	0
4	2	2	1	1	0	0	0	1	0	1	0	1	1	1	0
5			1	1	1	1	1	0	0	0	0	1	1	1	0
8	5	1	0	1	1	0	0	1	1	1	0	1	1	1	0
복원			**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**
원본			**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**0**

CASE 3: $W_{\min} = 3$ ($\theta = 0.1875$, $t = 6$)

Digit independent 알고리즘에 의한 원본데이터의 분할:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	0	0	1	0	0	0	0	1	0	1	0	1	0
2			1	1	1	1	0	1	0	1	0	1	1	1	0
3			1	1	0	0	0	0	1	0	0	1	0	1	1
4	2	2	0	0	1	0	1	1	0	1	0	0	1	0	1
5			1	1	1	0	0	1	0	1	1	1	1	1	0
6	3	1	1	1	1	0	1	1	1	1	1	1	1	0	0
7	4	1	1	1	0	0	0	1	0	1	0	0	1	1	0
8	5	1	1	1	1	1	0	1	0	0	0	1	1	1	0

[3-1] Group 1 's data(weight=3) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
4	2	2	0	0	1	0	1	1	0	1	0	0	1	0	1
5			1	1	1	0	0	1	0	1	1	1	1	1	0
6	3	1	1	1	1	1	0	1	1	1	1	1	1	0	0
7	4	1	1	1	1	0	0	0	1	0	1	0	0	1	1	0
8	5	1	1	1	1	1	1	0	1	0	0	0	1	1	1	0
복원			1	1	1	0	0	1	0	1	0	1	1	1	0
원본			1	1	1	0	0	1	0	1	0	1	1	1	0

[3-2] Group 2 & 3 's data(weight=3) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	0	0	1	0	0	0	0	1	0	1	0	1	0
2			1	1	1	1	0	1	0	1	0	1	1	1	0
3			1	1	0	0	0	0	1	0	0	1	0	1	1
7	4	1	1	1	1	0	0	0	1	0	1	0	0	1	1	0
8	5	1	1	1	1	1	1	0	1	0	0	0	1	1	1	0
복원			1	1	1	0	0	1	0	1	0	1	1	1	0
원본			1	1	1	0	0	1	0	1	0	1	1	1	0

[3-3] Group 3 & 4 & 5 's data(weight=3) was lost:

	G	W	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13
1	1	3	0	0	1	0	0	0	0	1	0	1	0	1	0
2			1	1	1	1	0	1	0	1	0	1	1	1	0
3			1	1	0	0	0	0	1	0	0	1	0	1	1
4	2	2	0	0	1	0	1	1	0	1	0	0	1	0	1
5			1	1	1	0	0	1	0	1	1	1	1	1	0
복원			1	1	1	0	0	1	0	1	0	1	1	1	0
원본			1	1	1	0	0	1	0	1	0	1	1	1	0

송 유 진

e-mail : song@dongguk.ac.kr
1982년 한국항공대학교(학사)
1987년 경북대학교 대학원(석사)
1995년 일본 Tokyo Institute of Technology(박사)
1988년~1996년 한국전자통신연구원 선임연구원
2003년~2005년 미국 University of North Carolina at Charlotte 연구교수
2006년 7월~8월 일본 정보보호대학원대학 객원교수
1996년~현 재 동국대학교 정보경영학과 교수
2005년~현 재 동국대학교 부설 전자상거래연구소 소장
1998년~현 재 한국정보보호학회 이사
2006년~현 재 국제e-비즈니스학회 이사
2006년~현 재 한국사이버테러정보전학회 이사
2001년 ICISC2001 운영위원장
2003년 하계CISC2003 프로그램위원장
2006년 CISC-S2006 공동프로그램 위원장
2007년 한국정보시스템학회 추계학술발표대회 공동 조직 위원장
관심분야 : Privacy Protection, Secret Sharing, 전자상거래응용 보안(Location Privacy, 디지털컨텐츠 보호, SCM/CRM 보안 등), Context Aware Application Security

김 진 석

e-mail : jinseog.kim@gmail.com
1992년 서울대학교 계산통계학과(학사)
1994년 서울대학교 대학원 통계학과 (이학석사)
2003년 서울대학교 대학원 통계학과 (이학박사)
2007년~현 재 동국대학교 정보통계학과 조교수
2003년~2007년 서울대학교 통계학과 Post-Doc.
관심분야 : 데이터마이닝, 기계학습, 정보보호 등