Ataque de reconstrucción de base de datos

Una persona realiza una serie de preguntas a una base de datos para investigar que hay en ella. Son ataques de inferencia para saber que hay dentro de una base de datos que ha sido curada. Utilizando la base de datos curada se trata de obtener datos sobre la base de datos original. Se recomienda evitar tener demasiados detalles en la versión curada.

Preguntas de un adversario

- Una posibilidad es denegar respuestas que podrían ser peligrosas
- El problema es que las denegaciones pueden filtrar información.

Censo ficticio del censo de EEUU

Se construyó una base de datos falsa con 7 personas simulando la del censo. Algunos datos como la edad han sido suprimidos para ciertas personas con el objetivo de proteger contra ataques de inferencia al haber demasiada poca gente cuyos datos como estos coinciden. A pesar de estar estos datos eliminados, se da acceso a datos estadísticos como la media y la mediana, lo que permite ir induciendo poco a poco las edades que han sido ocultadas. Con todo esto se puede proponer un sistema de inecuaciones que se puede aplicar a un algoritmo solver, resultando en que se pueden obtener los datos ocultos de esta forma. Como resultado de esto, el censo hizo un ataque de reconstrucción a su base de datos en 2010 utilizando los mismos principios y fueron capaces de reconstruir un 46% de la base de datos. Si se tolerara un pequeño margen de error se habría reconstruido un 71% de la base de datos. En total se pudieron identificar 50 millones de personas. A raíz de esto el censo comenzó a utilizar privacidad diferencial en 2020 que consiste en añadir ruido a las respuestas para dificultar este tipo de ataques.

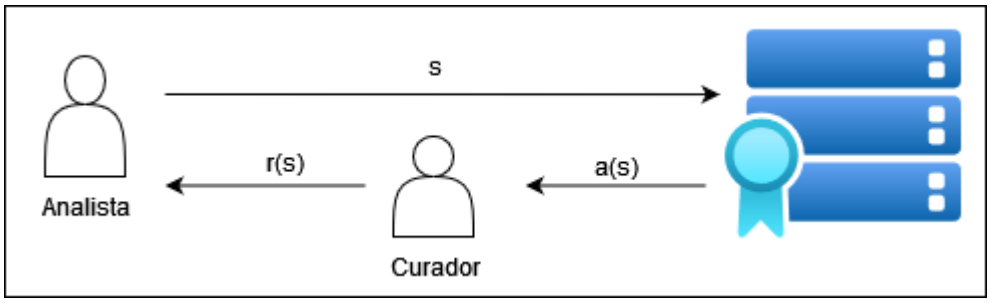
Modelo de base de datos para peticiones de subset

Un cuasi identificador es un conjunto de atributos que con información externa permiten identificar a una persona externa. Tomamos como ejemplo una tabla con n componentes que contiene varias columnas con nombre, id, codigo zip cumpleaños, género y una pregunta sensible. La pregunta sensible sería el secreto.

- n: numero de filas de una base de datos
- d∈{0,1}^n: vector secreto
- di: secreto en la fila i
- Vector de petición: s{0,1}^n tiene la misma estructura que el vector secreto, especifica un subset de filas que tienen cierto valor secreto.
- La respuesta a la petición s: a(s)
- Contador de preticiones: producto escalar entre s y d.

Curando respuestas

Responder a una petición s con la respuesta correcta a(s) viola la privacidad, si el adversario quiere encontrar el secreto para una fila en particular, solo debe construir el vector de petición con un 1 en la file y 0 en el resto de sitios. Por lo que la respuesta r(s) debe ser una versión ruidosa o aleatorizada de a(s). Entre la respuesta correcta y la respuesta real, como mucho debe haber cierto número determinado de diferencia.



Ataque de reconstrucción lineal

Si el analista puede llegar a hacer 2^n preguntas y el curador añade ruido con un límite E, entonces el analista es capaz de reconstruir la base de datos con la excepción de $4E$ posiciones.

1. Primero se envían todas las solicitudes posibles y se almacenan las respuestas de la base de datos (El adversario colecciona todas las posibles respuestas)
2. Tras eso se encuentra una lista de candidatos válidos, esta lista suele ser bastante pequeña, el vector secreto va a estar en esta lista de candidatos.
3. Hay una propiedad tal que c^(k^(-1))

Corolario: a menos que haya un limite en las peticiones de la base de datos, una reconstrucción casi perfecta es posible dentro de 4E posiciones. Por lo que es posible reconstruir la base de datos hasta el 99% de las posiciones.

A la hora de la verdad esto se puede hacer con un número de peticiones mucho más reducido por lo Aunque el ruido que se añada esté acotado como $E = √(a√n)$, si se le deja al atacante hacer peticiones de orden n, entonces es capaz de reconstruir la base de datos en casi todas las posiciones. Este ataque utiliza programación lineal, que es un ataque estadístico.

Desafío Aircloak's Diffix

La compañía Aircloak, que vendía el producto Diffix, prometía privacidad en base a curar las respuestas. Puso un concurso de 5000$ para ver si su producto respondía a los ataques de reconstrucción como se esperaba. La cantidad de ruido que se añadía a cada respuesta era de la raiz cuadrada del número de condiciones. Estaban prohibidas operaciones de tipo OR en la base de datos y en caso de que la salida tuviera pocas entradas la base de datos no respondería. Se planteó una forma de interrogar la base de datos de forma que las peticiones cubrieran muchos elementos de la base de datos de una forma más o menos aleatoria y que tuvieran un número de condiciones muy pequeño. Como resultado se obtuvo la siguiente consulta SQL:

```
SELECT COUNT(clientId)
FROM loans
WHERE FLOOR(100*((clientId*2)^0,7)+0.5) = FLOOR(100*(clientId*2)^0.7)
AND clientID BETWEEN 2000 AND 3000
AND loanStatus = 'C'
```

Con esta consulta se pudo reconstruir la base de datos a pesar de las medidas tomadas por Aircloak.