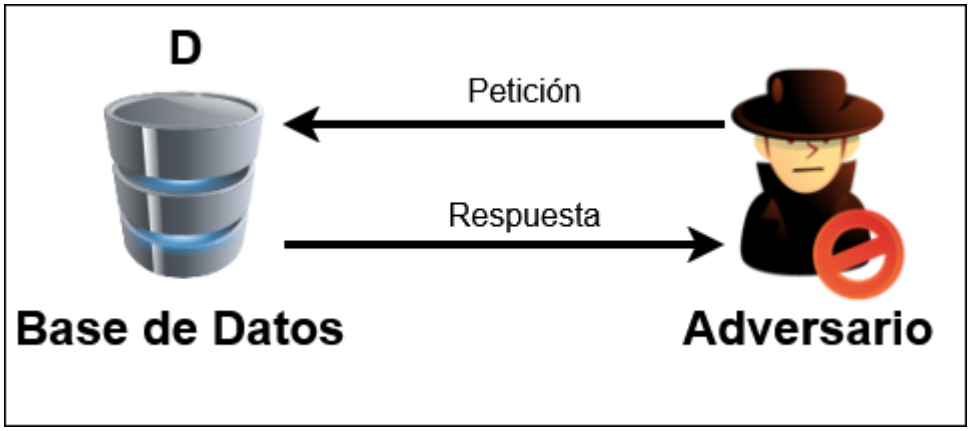
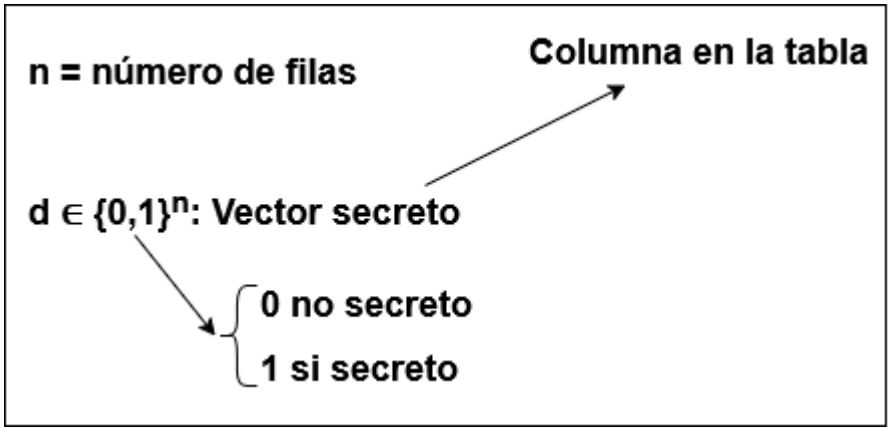
Ataques de reconstrucción de Bases de Datos

Un adversario quiere inferir algo de una base de datos mediante peticiones. Un ataque de inferencia está pensado para reconstruir una base de datos previamente curada. Se recomienda curar la base de datos evitando que esta tenga datos demasiado detallados.



Queremos proteger la base de datas contra el exceso de inferencias. El adversario puede tener una estrategia de peticiones dinámicas que van cambiando en fuinción a la información que van obteniendo. Para proteger los datos, se pueden denegar respuestas en función a lo peligrosas que puedan ser con respecto a la privacidad. El problema de denegar información es que a su vez puede dar información sin querer al adversario. Por ejemplo, si se pide el mayyor sueldo de un rango de empleados, sale X cifra, si luego hacemos la misma petición quitando un empleado y aparece una cifra diferente, podemos inferir el sueldo de dicho empleado.



El atacante hace preguntas usando un vector que tiene los datos tratando de descubrir si un dato es secreto o no.

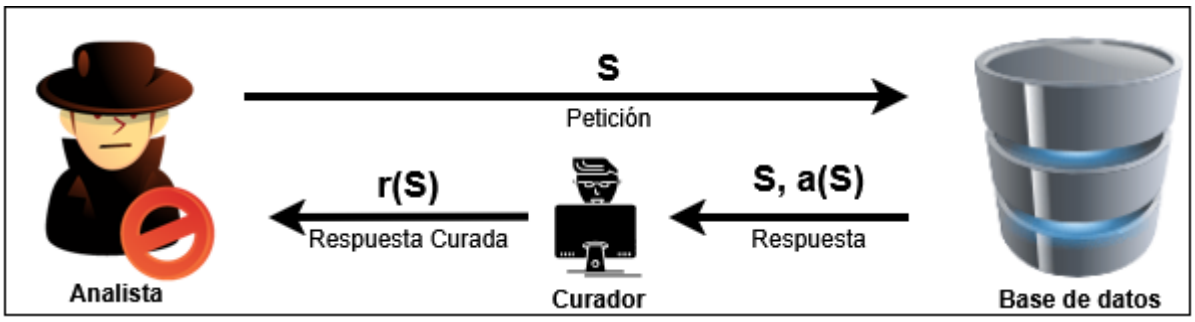
Censo Ficticio de EEUU

Se construyó una base de datos falsa simulando el censo de EEUU. 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 con datos que coincidan con ellos. 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 en dicha Base de Datos. Como resultado, el censo hizo un ataque de reconstrucción a su base de datos en 2010 usando estos mismos principios, resultando en la reconstrucción de un 46% de la base de datos sin errores y un 71% de la base de datos tolerando algunos errores. En total se pudo identificar a 50 millones de personas. A raíz de esto el censo comenzó a usar privacidad diferencial en 2020, añadiendo ruido a las respuestas para dificultar ataques de reconstrucción.

Curando Respuestas

Responder a una petición S con la respuesta verdadera a(S) viola la privacidad, si el adversario quiere descubrir un dato oculto para cierta fila, simplemenete tiene que construir un vector de peticiones con un 1 en la fila y 0 en el resto. Para evitar violar la privacidad la respuesta r(S) debe ser una versión de a(S) con ruido añadido. Para preservar la utilidad de los datos se deben estableecer unos límites de distorsión como, por ejemplo, $|r(s) - a(s)| ≤ E$

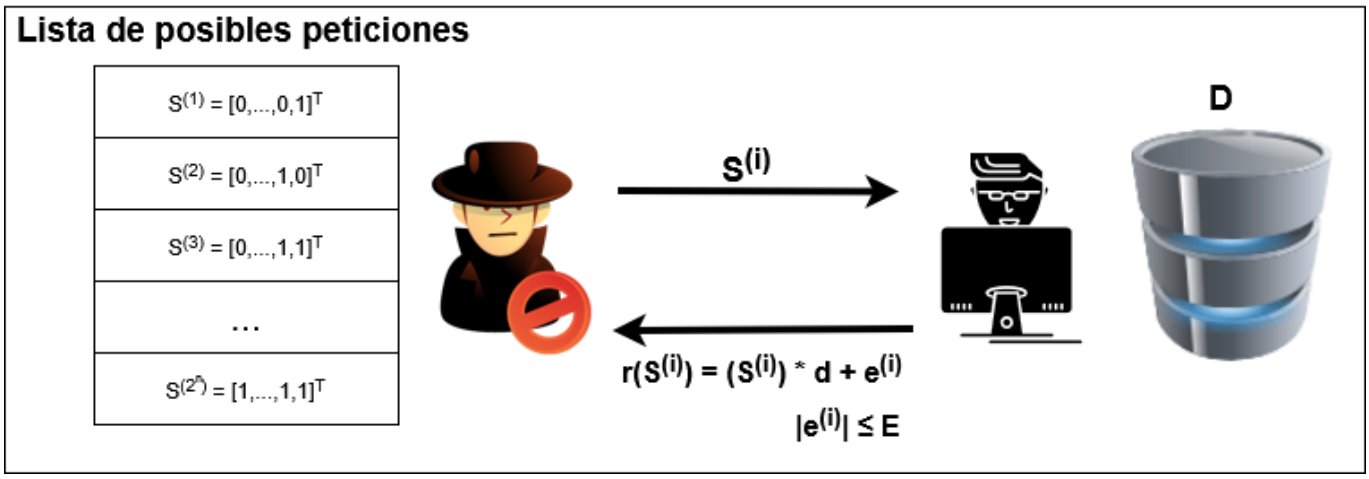


Ataque de reconstrucción lineal

Teorema: Si el analista puede realizar 2^n peticiones y el curador añade ruido E, entonces el adversario puede reconstruir la base de datos entera salvo por $4E$ posiciones.

Fase 1

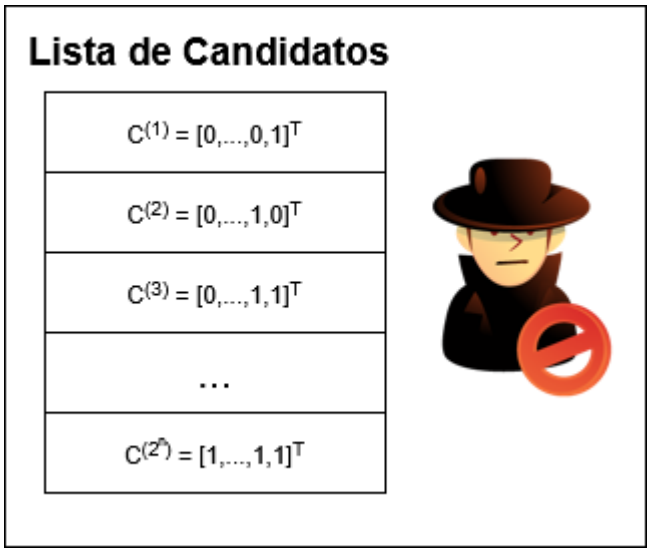
- El analista envía todas las peticiones posibles a la base de datos y almacena las respuestas.




```
for i = 1:2^n
    Analista envía s(i)
    Curador responde con r(s(i)) ) s(i)^T * d + e(i)
```

Fase 2

- El analista busca candidatos válidos




```
for k = 1:2^n
  candidate = 'valid'
  for i = 1:2^n
    if |r(s(i)) - (s(i))^T * c(k)|
> E:
      canditate = 'not valid'
      break
  if candidate = 'valid' : store
c(k) in lista de candidatos válidos
```

From:
https://knoppia.net/ - **Knoppia**

Permanent link:
https://knoppia.net/doku.php?id=pan:ataques_recontruccion_v2&rev=1767540962

Last update: **2026/01/04 15:36**