Criptografía homomórfica: calculando en el ciberespacio

Elvira Mayordomo

Universidad de Zaragoza

29 de octubre de 2012

Objetivo de la criptografía homomórfica

- **Calular con datos que no se pueden ver**
- ¿Cómo puede una compañía externalizar cálculos con datos que quiere mantener secretos?
- Alquila tiempo de cálculo a otra compañía y le manda sus datos encriptados
- ¿Es posible que la otra compañía pueda hacer los cálculos sin conocer los datos (y sin conocer el resultado)?

Objetivo de la criptografía homomórfica

- Se trata de intentar hacer criptografía homomórfica, queremos realizar el cálculo *calc* con los datos $m_1, \ldots, m_k$
- Queremos, a partir de

$$e(m_1, k), \ldots, e(m_r, k)$$

calcular

$$e(calc(m_1, \ldots, m_r), k)$$

- *calc* es un programa cualquiera, siempre se puede descomponer en una serie de operaciones sencillas (sumas y multiplicaciones)
- Por ejemplo RSA puede hacerlo con multiplicaciones, a partir de $e(m_1, k), \ldots, e(m_r, k)$ es fácil calcular $e(m_1 * \ldots * m_r, k)$

Resultado rompedor en 2009

- Durante mucho tiempo no se consiguió, se podían hacer unas pocas operaciones pero los errores aumentaban muy rápido
- Gentry lo consigue en 2009 utilizando:
  - Además de $e(m_1, k), \ldots, e(m_r, k)$ se dispone de $e(k, k)$
  - Con ello de vez en cuando se "refrescan" los datos mejorando el error que se va acumulando
- Actualmente se trabaja en los problemas de eficiencia

Dominios de aplicación

- Historiales médicos: un paciente le da acceso a su médico sólo a los datos extrictamente necesarios (haz los cálculos que necesites y yo te decodifico el resultado)
- Muro anunciantes-consumidores: mandar los anuncios a los consumidores interesados sin que los anunciantes tengan acceso a sus datos
- Filtros de spam para mensajes encriptados: un filtro de spam que puede actuar sobre mensajes encriptados con mi clave pública sin poder leerlos
- Un banco homomórfico: ni siquiera los empleados conocen las transacciones: realizan operaciones sin conocer los detalles