1 Problem formulation

Let D be a two-dimensional table that supports the following operations:

- **Insert:** add a new row to the table.
- **Delete:** remove a row from the table.
- **Lookup:** find rows in the table that contains some keyword given as the input to the *lookup* function.

Further, we assume that D has n columns, with S_i the set possible attributes in the i-th column. We call D a database.

Our goal is to construct a cryptographic database D that is secure when outsourced: no dishonest third-party server should be able to decrypt the database. We also want the database to be efficient on the operations above. In particular, we want *lookup(·)* to be sub-linear time.

2 Constructions

Without loss of generality, we assume that D has $n-1$ columns of actual entries. The n-th column is an auxiliary column that indicates if the corresponding row is genuine or fake.

The message to be encrypted is denoted as $m = (m_1, m_2, ..., m_{n-1})$. So m_i is the plaintext for the i-th column for this particular message. As a short hand, we write $\mathrm{Enc}(m, \mathsf{pk}) = (\mathrm{Enc}(m_1, \mathsf{pk}), \mathrm{Enc}(m_2, \mathsf{pk}), ..., \mathrm{Enc}(m_{n-1}, \mathsf{pk}))$ to be the encryption scheme Enc applied to the message under public key pk.

We write (D, C) to mean insertion of C (as rows) into the database D, and $C \| x$ to mean concatenation of column x to C.

For the constructions below, we encrypt the first $n-1$ columns deterministically. The auxiliary column is encrypted using a probabilistic encryption scheme.

Let DE = $(\mathrm{Kg}_1, \mathrm{Enc}_1, \mathrm{Dec}_1)$ be the deterministic encryption scheme and PKE $= (\mathrm{Kg}_2, \mathrm{Enc}_2, \mathrm{Dec}_2)$ be the probabilistic encryption scheme. Let $rand(C)$ be a function that shuffles rows of C. We define the following encryption schemes for databases.

2.1 Exponential-space construction

Key Generation(1^n)

$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathrm{Kg}_1(1^n)$
$(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathrm{Kg}_2(1^n)$
$\mathsf{pk} \leftarrow (\mathsf{pk}_1, \mathsf{pk}_2)$
$\mathsf{sk} \leftarrow (\mathsf{sk}_1, \mathsf{sk}_2)$

Insert(m)

$(\mathsf{pk}_1, \mathsf{pk}_2) \leftarrow \mathsf{pk}$
for $x \in S_1 \times S_2 \times ... \times S_{n-1}$
  if $x = m$
    $D \leftarrow (D, (\mathrm{Enc}_1(m, \mathsf{pk}_1) \| \mathrm{Enc}_2(True, \mathsf{pk}_2)))$
  else
    $D \leftarrow (D, (\mathrm{Enc}_1(x, \mathsf{pk}_1) \| \mathrm{Enc}_2(False, \mathsf{pk}_2)))$

Decrypt(D)

$(\mathsf{sk}_1, \mathsf{sk}_2) \leftarrow \mathsf{sk}$
$m \leftarrow ()$
for c in D
  parse c as $\bar{c}||x$
  if $\mathrm{Dec}_2(x, \mathsf{sk}_2) = True$
    $m \leftarrow (m, \mathrm{Dec}_1(c))$
return m

lookup(c, i)

$r \leftarrow ()$
for c in D
  if $c_i = c$
    $r \leftarrow (r, c)$
return r

3 Security notions

3.1 Indistinguishability of distributions

3.2 Indistinguishability of plaintext (1)

3.3 Indistinguishability of plaintext (2)