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 $\text{Enc}(m, \mathsf{pk}) = (\text{Enc}(m_1, \mathsf{pk}), \text{Enc}(m_2, \mathsf{pk}), ..., \text{Enc}(m_{n-1}, \mathsf{pk}))$ 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 $= (\text{Kg}_1, \text{Enc}_1, \text{Dec}_1)$ be the deterministic encryption scheme and PKE $= (\text{Kg}_2, \text{Enc}_2, \text{Dec}_2)$ 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

One way to define security notion for deterministic encryption on databases is to consider indistinguishability of input distributions. Suppose that we tell the adversary that the underlying plaintext is from one of the two distributions, he should not be able to tell which one of the distribution the database is from with significant advantage.

Let Π_0 and Π_1 be two distinct streams of the plaintext of equal size, and b $\leftarrow_{\$} \{0, 1\}$ by the challenger, we establish the following game:

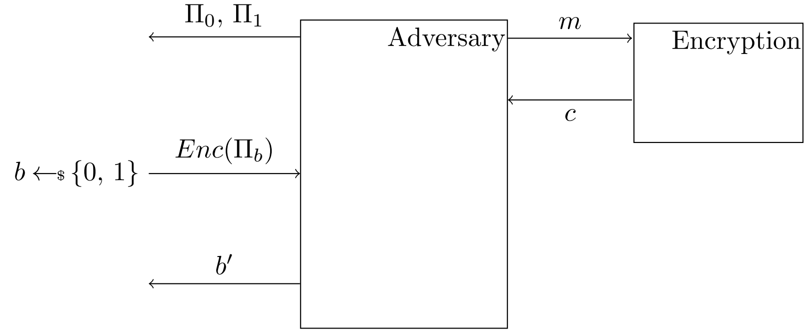


Figure 1: IND of distributions

Description of the game:

1. The adversary and the challenger agree on the set of plaintexts.
2. The adversary generates two streams of plaintexts of equal length and sends them to the challenger.
3. The challenger randomly samples the guess bit b from $\{0, 1\}$ and encrypts Π_b using some encryption algorithm. The result is sent back to the adversary.
4. The adversary has to guess if the encryption comes from Π_0 or Π_1 and outputs his guess bit b' (in polynomial time). For a CPA adversary, he has access to an encryption oracle that can encrypt any message he wants other than the set of plaintexts he agrees with the challenger. In case of a CCA adversary, he also has access to a decryption oracle.

3.2 Unidentifiability of plaintext

One advantage of using DE is that when making queries (to third-party database server), one does not need to disclose the underlying plaintext. So we need to make sure that the database itself does not leak any information about the underlying plaintext.

In terms of security goal, DE on databases operates on a very different environment as compared to classic cryptographic protocols. There is no reason to assume that the adversary does not know about the set of underlying plaintext, and if he does, the minimal probability of him making a right guess of a plaintext-ciphertext pair is one over the number of plaintexts. In this security model, we require the success probability of the adversary in guessing the right plaintext-ciphertext pair to not be significantly larger than one over the number of plaintexts.

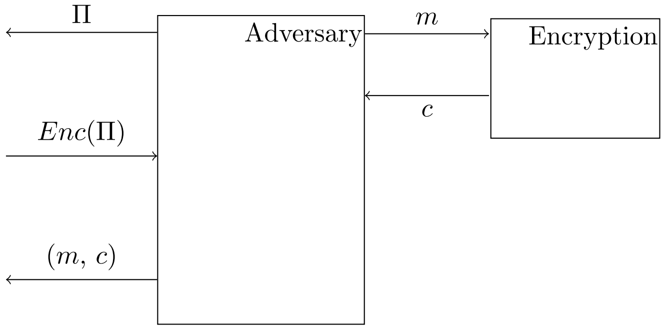


Figure 2: Unidentifiability of plaintext

Description of the game:

1. The adversary and the challenger agree on the set of plaintexts.
2. The adversary generates a stream of plaintexts and send it to the challenger.
3. The challenger encrypts the plaintexts with some encryption scheme and send it to the adversary.
4. The goal of the adversary is to return a valid pair of plaintext and ciphertext in polynomial time. For a CPA adversary, he has access to an encryption oracle that can encrypt any message he wants other than the set of plaintexts he agrees with the challenger. In case of a CCA adversary, he also has access to a decryption oracle.

3.3 Indistinguishability of plaintext

On a lower level, we want to analyse IND property of individual plaintexts. This security definition is very similar to the classical IND game except that the encryption is no longer a single ciphertext. Similar to the standard IND game, b is randomly generated from $\{0, 1\}$ by the challenger.

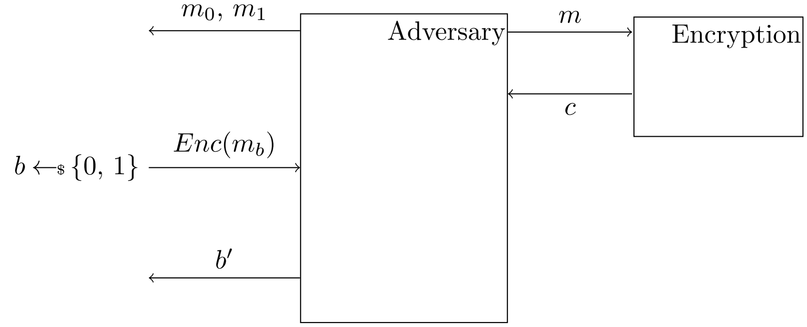


Figure 3: IND of plaintext

4 Security proofs

4.1 Unidentifiability of plaintext

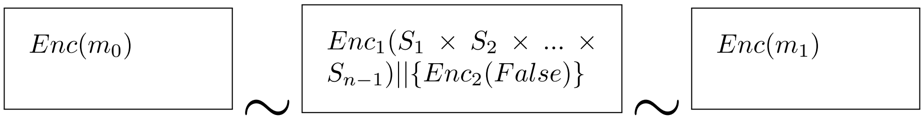
4.2 Indistinguishability of plaintext

Suppose now that we are working with the deterministic padding scheme. We wish to prove that the scheme is secure under IND of plaintext. Let us denote the adversary by A, who's input is encryption of m_b, then the scheme is secure if and only if the advantage:

$$\mathsf{Adv}_{\mathcal{A}}(n) = |\Pr_{b \leftarrow 0}[A(Enc_{\mathsf{pk}}(m_b)) = 0] - \Pr_{b \leftarrow 1}[A(Enc_{\mathsf{pk}}(m_b)) = 0]|$$

is negligible for all adversaries A and public keys pk.

This IND is computational in nature and it is no different from PRGs, so we are essentially trying to prove that $Enc_{\mathsf{pk}}(m_0) \sim Enc_{\mathsf{pk}}(m_1)$. This can be proven with a hybrid argument:


$Enc(m_0)$
$\sim$
$Enc_1(S_1 \times S_2 \times ... \times S_{n-1})||\{Enc_2(False)\}$
$\sim$
$Enc(m_1)$

(1) Encryption of message m_0

(2) Encryption of message m_0 with its auxiliary column set to false

(3) Encryption of message m_1

Figure 4: Hybrid structure of the proof

In the figure above, we have omitted the public keys in the notation. By $Enc_1(S_1 \times S_2 \times ... \times S_{n-1}) || \{Enc_2(False)\}$, we really mean the set

$$\{x || Enc_2(False) \mid x \in S_1 \times S_2 \times ... \times S_{n-1}\},$$

where each row of the auxiliary column receives a fresh encryption of the flag.

Proof of (1) $\sim$ (2): By definition, the first n-1 columns of $Enc(m_0)$ are identical to that of (2) so distinguishability relies completely on the auxiliary column. We claim that if the underlying encryption scheme $Enc_2(\cdot)$ is IND-CPA (or IND-CCA) secure, then the two auxiliary columns are indistinguishable.

Proof of the claim: We prove the claim by reduction. Suppose otherwise that we can distinguish the two auxiliary columns with a distinguisher D (which outputs 0 if the column contains an encryption of $True$), and our goal is to break the security of $Enc_2(\cdot)$. We establish the adversary A against $Enc_2(\cdot)$ as follows:

```
Adversary $A$
$m_0 \leftarrow True$
$m_1 \leftarrow False$
$b \leftarrow_{\$} \{0, 1\}$
$c \leftarrow Enc_2(m_b)$
$L \leftarrow (c)$
for $i = 1... |S_1 \times S_2, ... \times S_{n-1}|$ :
    $L \leftarrow (L, Enc_2(False))$
$b' \leftarrow D(L)$
return $b'$
```

Figure 5: Adversary A against $Enc_2(\cdot)$

It is straightforward that $\Pr_{b \leftarrow 0}[0 \leftarrow A] = \Pr_{b \leftarrow 0}[0 \leftarrow D(L)]$ and $\Pr_{b \leftarrow 1}[0 \leftarrow A] = \Pr_{b \leftarrow 1}[0 \leftarrow D(L)]$. In particular, the advantage of A is

$$|\Pr_{b \leftarrow 0}[0 \leftarrow A] - \Pr_{b \leftarrow 1}[0 \leftarrow A]| = |\Pr_{b \leftarrow 0}[0 \leftarrow D(L)] - \Pr_{b \leftarrow 1}[0 \leftarrow D(L)]|$$

which is significant by our assumption. So IND-CPA security of $Enc_2(\cdot)$ guarantees the indistinguishability of (1) and (2).

Proof of (2) $\sim$ (3): The prove above uses arbitrary m_0 so the statement is true for all messages. In particular, it is true for m_1. Whence, we conclude that (2) $\sim$ (3).

Conclusion: By combining the results above, we have shown that (1) $\sim$ (2) $\sim$ (3). Thus, (1) $\sim$ (3). □

On a side note, the proof above has not used any property of $Enc_1(\cdot)$. In fact, this security notion is valid even if $Enc_1(\cdot) = id(\cdot)$, i.e. we do not encrypt the plaintext at all.

It is important to note that IND of plaintext does not imply IND of distribution. Specifically, the naive DE satisfies this definition of security but it is not secure under IND of distributions, as demonstrated by frequency attack.

4.3 Indistinguishability of distributions

Indistinguishability of distributions can be proven the exact same way as of figure 4 for the full padding scheme.

Proof: Suppose that Π_0 and Π_1 are two streams of input plaintext with length k, we adapt the hybrid argument as:

$$\boxed{Enc(\Pi_0)} \sim \boxed{(Enc_1(S_1 \times S_2 \times ... \times S_{n-1})||\{Enc_2(False)\})^k} \sim \boxed{Enc(\Pi_1)}$$

(1) Encryption of plaintext stream Π_0

(2) Encryption of plaintext stream Π_0 with its auxiliary column set to false

(3) Encryption of plaintext stream Π_1

Figure 6: Hybrid structure of the proof

In the notation, we write $(Enc_1(S_1 \times S_2 \times ... \times S_{n-1})||\{Enc_2(False)\})^k$ for k copies of $Enc_1(S_1 \times S_2 \times ... \times S_{n-1})||\{Enc_2(False)\}$ with fresh encryptions of *False* for all rows as a joint database.

Similar to the proof of IND of plaintext, the first n-1 columns of (1) and (2) are identical so IND property is completely determined by the auxiliary column. By construction, the auxiliary column can be divided into blocks of size $|S_1 \times S_2 \times ... \times S_{n-1}|$ each, corresponding to each of the messages in the plaintext stream. Because we already know that IND-CPA security of $Enc_2(\cdot)$ implies IND of the auxiliary columns for single messages, it suffices to prove that the later implies IND of the auxiliary columns for any message stream.

Proof of the claim: We proceed by reduction. Suppose otherwise that the auxiliary columns for messages streams can be distinguished, and suppose D is a distinguisher, we want to show that D can be used to construct an adversary

A against the auxiliary column for single messages. But by construction, the message stream can be a single message, so the distinguisher D can be used directly to construct A:

```
Adversary $A(L)$
$b' \leftarrow D(L)$
return $b'$
```

Figure 7: Adversary A against auxiliary column for single messages

Thus, the probability of A succeeding equals to that of D, which is clearly not negligible. This proves the claim as desired.

Further, our choice of message stream Π_0 in the proof is arbitrary, so the statement holds for all message streams (of the same length as Π_0). Hence, we conclude that (2) $\sim$ (3). Therefore, (1) $\sim$ (3) as required by the security notion. □