Security of Deterministic Encryption

Zichen Gui
Supervised by: Dr. Bogdan Warinschi

February 21, 2017

Outline of the presentation

- Introduce to encryption schemes with additional properties
- Describe deterministic encryption scheme
- Demonstrate usefulness of deterministic encryption
- Analyse the security of deterministic encryption
- Show that deterministic encryption is not secure in database applications
- Discuss the approaches of the research

Introduction

- Security of traditional encryption schemes are based on indistinguishability (IND) of ciphertexts or semantic security
- In particular, they need to be probabilistic
- Not efficient for processing encrypted data

Introduction

- Schemes allowing for better processing of encrypted data often process some additional properties
- Usual security notion of IND cannot be satisfied
- Need to understand the maximal level of security those schemes can offer

Construction of deterministic encryption (1)

- Based on a (secure) hash function H and a probabilistic encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$

Key generation

$(\mathrm{pk}, \mathrm{sk}) \leftarrow_{\$} \mathcal{K}(1^n)$

Encryption(pk, x)

$\omega \leftarrow H(\mathrm{pk} \| x)$
$y \leftarrow \mathcal{E}(\mathrm{pk}, x; \omega)$
return y

Decryption(pk, sk, y)

$x \leftarrow \mathcal{D}(\mathrm{sk}, y)$
$\omega \leftarrow H(\mathrm{pk} \| x)$
if $\mathcal{E}(\mathrm{pk}, x; \omega) = y$
    return x
return $\perp$

Figure: Deterministic encryption based on hashing

Construction of deterministic encryption (2)

- Use trapdoor permutation in place of hash function
- Trapdoor permutation is a family of functions that is easy to compute but hard to invert
- If given some additional information, known as the 'trapdoor', the inversion can be computed efficiently

Construction of deterministic encryption (2)

Key generation	Encryption(pk, x)
1 : $(\phi, \tau) \leftarrow_{\$} \mathcal{G}(1^n)$	1 : $(\phi, \bar{\mathrm{pk}}, p) \leftarrow \mathrm{pk}$
2 : $s \leftarrow_{\$} \{0,1\}^n$	2 : $y \leftarrow F(\phi, x)$
3 : $(\bar{\mathrm{pk}}, \bar{\mathrm{sk}}) \leftarrow_{\$} \mathcal{K}(1^n)$	3 : $\omega \leftarrow \mathit{GetCoins}(F, \phi, x, s)$
4 : $\mathrm{pk} \leftarrow (\phi, \bar{\mathrm{pk}}, s)$	4 : $c \leftarrow \mathcal{E}(\mathrm{pk}, y; \omega)$
5 : $\mathrm{sk} \leftarrow (\tau, \bar{\mathrm{sk}})$	5 : **return** c
6 : **return** $(\mathrm{pk}, \mathrm{sk})$	

Figure: Deterministic encryption based on trapdoor permutations

Construction of deterministic encryption (2)

$$
\begin{array}{ll}
\underline{\text{Decryption}(\mathsf{pk},\ \mathsf{sk},\ y)} & \\
1: & (\tau, \bar{\mathsf{sk}}) \leftarrow \mathsf{sk} \\
2: & y \leftarrow \mathcal{D}(\bar{\mathsf{sk}}, c) \\
3: & x \leftarrow \bar{F}(\tau, y) \\
4: & \textbf{return } x
\end{array}
$$

Figure: Deterministic encryption based on trapdoor permutations

Usefulness of Deterministic Encryption

- Overcomes poor source of randomness used in probabilistic encryption schemes
- Allows for efficient searching in databases:
  - log-time with binary trees such as red-black tree or 2,3,4-tree
  - log-log-time with Van Emde Boas tree

- An IND adversary is a triple $I = (I_c, I_m, I_g)$ of PPT algorithms
- I_c: generates a state that will be used by I_m
- I_m: generates a pair of messages given the state
- I_g: guess the challenge bit b

Security of Deterministic Encryption

Original definition

$$\text{Experiment } \mathsf{EXP}_{\mathsf{I}}^{\mathsf{IND}}(n)$$

1 : $st \leftarrow I_c(1^n)$

2 : $(m_0, m_1) \leftarrow I_m(st)$

3 : $b \leftarrow_\$ \{0, 1\}$

4 : $c \leftarrow \text{Encryption}(pk, m_b)$

5 : $b^{'} \leftarrow I_g(\text{pk}, c, st)$

Figure: IND game for deterministic encryption

Security of Deterministic Encryption

Original definition

Differences to the standard IND security:

- The algorithms I_c and I_m accessed by the adversary have no access to the public key.
- The final algorithm I_g only has access to the state generated by I_c, instead of the plaintexts (m_0, m_1).
- The message space has high min-entropy.

Security of Deterministic Encryption

Alternative definition

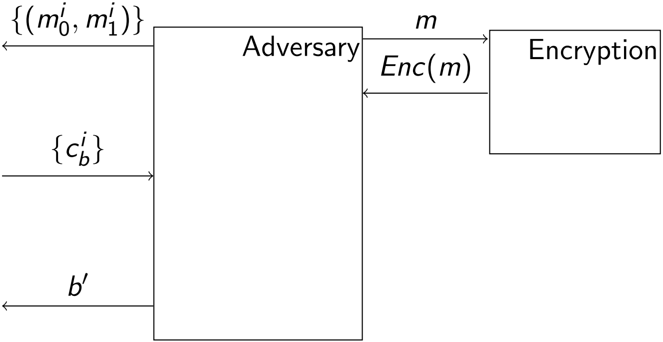

Figure: Cryptographic game of IND-DCPA

Attack on deterministic encryption in databases

- Deterministic encryption leaks frequency
- Frequency of ciphertexts can be compared to auxiliary data or prior knowledge to match the ciphertexts to plaintexts

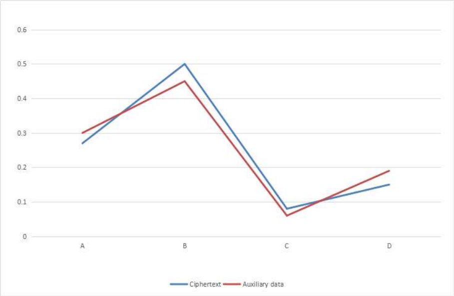

Figure: Attack by comparing frequency of some auxiliary data and ciphertexts

Summary

In the presentation, we have discussed:

- Construction of deterministic encryption (DE)
- Usefulness of DE in database applications
- DE is not secure in database applications

Discussion

Scientific questions to be addressed in the project:

- How should security notion be defined for DE in application to databases?
- Is there an encryption scheme that is secure under that security definition?