

[Google Discussions](#)

### Exam Professional Cloud Security Engineer All Questions

View all questions & answers for the Professional Cloud Security Engineer exam

[Go to Exam](#)

## EXAM PROFESSIONAL CLOUD SECURITY ENGINEER TOPIC 1 QUESTION 101 DISCUSSION

Actual exam question from Google's Professional Cloud Security Engineer

Question #: 101

Topic #: 1

[\[All Professional Cloud Security Engineer Questions\]](#)

Your company wants to determine what products they can build to help customers improve their credit scores depending on their age range. To achieve this, you need to join user information in the company's banking app with customers' credit score data received from a third party. While using this raw data will allow you to complete this task, it exposes sensitive data, which could be propagated into new systems.

This risk needs to be addressed using de-identification and tokenization with Cloud Data Loss Prevention while maintaining the referential integrity across the database. Which cryptographic token format should you use to meet these requirements?

- A. Deterministic encryption
- B. Secure, key-based hashes
- C. Format-preserving encryption
- D. Cryptographic hashing

[Show Suggested Answer](#)

by  gionny at April 26, 2022, 3:26 p.m.

### Comments

[Submit](#)

  **mT3** Highly Voted  3 years, 2 months ago

**Selected Answer: A**

"This encryption method is reversible, which helps to maintain referential integrity across your database and has no character-set limitations."



<https://cloud.google.com/blog/products/identity-security/take-charge-of-your-data-how-tokenization-makes-data-usable-without-sacrificing-privacy>

   upvoted 11 times

  **[Removed]** 2 years ago

I meant both A and C not A and D.

   upvoted 1 times

  **AzureDP900** 2 years, 8 months ago

A is right

   upvoted 1 times

  **YourFriendlyNeighborhoodSpider** Most Recent  4 months, 2 weeks ago

**Selected Answer: C**

C. Format-preserving encryption

Justification Based on Documentation:

[https://cloud.google.com/dlp/docs/transformations-reference#transformation\\_methods](https://cloud.google.com/dlp/docs/transformations-reference#transformation_methods)

According to the Google Cloud DLP guidelines, format-preserving encryption (FPE) transforms sensitive data while keeping its original format. This is essential for working with structured data where you need to maintain the integrity of data types (e.g., keeping a credit score as a numeric field) while ensuring security through encryption.

The ability to join user information in the banking app with credit score data while preserving the structure and format of the data is critical, especially since the goal is to analyze the data without exposing sensitive information.

   upvoted 2 times

  **BPzen** 8 months ago

**Selected Answer: C**

Why C. Format-preserving encryption is correct:

Format-preserving encryption (FPE) encrypts data while preserving its format (e.g., encrypting a credit card number would still result in a string with the same length and structure).

It ensures that data relationships and referential integrity across systems remain intact.

FPE is supported by Google Cloud DLP for tokenization tasks.

Why not the other options:

A. Deterministic encryption:

Deterministic encryption ensures that the same plaintext always encrypts to the same ciphertext, which can preserve referential integrity. However, it doesn't inherently maintain the format of the original data, which might be a requirement in this case.

   upvoted 2 times

  **YourFriendlyNeighborhoodSpider** 4 months, 2 weeks ago

YES, C is correct:

[https://cloud.google.com/dlp/docs/transformations-reference#transformation\\_methods](https://cloud.google.com/dlp/docs/transformations-reference#transformation_methods)

Format preserving encryption: Replaces an input value with a token that has been generated using format-preserving encryption (FPE) with the FFX mode of operation. This transformation method produces a token that is limited to the same alphabet as the input value and is the same length as the input value. FPE also supports re-identification given the original encryption key.

-> The key is that we talk about tokenization.

   upvoted 1 times

  **rsamant** 1 year, 7 months ago

D Cryptographic hashing as it maintains referential integrity and not reversible

<https://cloud.google.com/dlp/docs/pseudonymization>

   upvoted 3 times

  **Xoxoo** 1 year, 10 months ago

**Selected Answer: A**

To meet the requirements of de-identifying and tokenizing sensitive data while maintaining referential integrity across the database, you should use "Deterministic encryption."

Deterministic encryption is a form of encryption where the same input value consistently produces the same encrypted output (token). This ensures referential integrity because the same original value will always result in the same token, allowing you to link and join data across different systems or databases while still protecting sensitive information.

Format-preserving encryption is a specific form of deterministic encryption that preserves the format and length of the original data, which can be useful for maintaining data structures and relationships.

So, the correct option is:

A. Deterministic encryption

👍 🔄 🚩 upvoted 2 times

🗳️ 👤 **[Removed]** 2 years ago

**Selected Answer: A**

"A"

Requirements are reversible while maintaining referential integrity. Both A and D meet this requirement however D has input limitations. Therefore A is a better answer.

[https://cloud.google.com/dlp/docs/transformations-reference#transformation\\_methods](https://cloud.google.com/dlp/docs/transformations-reference#transformation_methods)

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **danidee111** 2 years, 1 month ago

This is a poor question and not enough data is provided to determine which Tokenization method should be selected. There are three methods for Tokenization (also referred to as Pseudonymization). See:

<https://cloud.google.com/dlp/docs/transformations-reference#crypto> and each method maintains referential integrity See: <https://www.youtube.com/watch?v=h0BnA7R8vg4>. Thus, you'd need to know whether it needs to be reversible, format preserving to confidentially select an answer..

👍 🔄 🚩 upvoted 3 times

🗳️ 👤 **gcpengineer** 2 years, 2 months ago

**Selected Answer: A**

<https://cloud.google.com/blog/products/identity-security/take-charge-of-your-data-how-tokenization-makes-data-usable-without-sacrificing-privacy>

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **passex** 2 years, 7 months ago

"Deterministic encryption" is too wide definition, the key phrase is "Which cryptographic token format " so th answer is "Format-preserving encryption" - where Referential integrity is assured (...allows for records to maintain their relationship ....ensures that connections between values (and, with structured data, records) are preserved, even across tables)

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **gcpengineer** 2 years, 2 months ago

A is the ans. <https://cloud.google.com/blog/products/identity-security/take-charge-of-your-data-how-tokenization-makes-data-usable-without-sacrificing-privacy>

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **PST21** 2 years, 7 months ago

Cryptographic uses strings , it asks to use tokenization and hence deterministic is better than FPE hence A

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **gcpengineer** 2 years, 2 months ago

both create tokens, the FPE is more used where u have format [0-9a-za-Z]

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **Littleivy** 2 years, 8 months ago

**Selected Answer: D**

Though it's not clear, but, to prevent from data leak, it's better to have a non-reversible method as analysts don't need re-identification

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **AwesomeGCP** 2 years, 9 months ago

**Selected Answer: A**

A. Deterministic encryption

👍 🔄 🚩 upvoted 1 times

🗳️ 👤 **zelck** 2 years, 10 months ago

**Selected Answer: A**

A is the answer.

<https://cloud.google.com/dlp/docs/pseudonymization>

FPE provides fewer security guarantees compared to other deterministic encryption methods such as AES-SIV.

For these reasons, Google strongly recommends using deterministic encryption with AES-SIV instead of FPE for all security sensitive use cases.

Other methods like deterministic encryption using AES-SIV provide these stronger security guarantees and are recommended for tokenization use cases unless length and character set preservation are strict requirements—for example, for backward compatibility with a legacy data system.

👍 🔄 🚩 upvoted 4 times

🗨️ 👤 **piyush\_1982** 3 years ago

**Selected Answer: A**

This question is taken from the exact scenario described in this link

<https://cloud.google.com/blog/products/identity-security/take-charge-of-your-data-how-tokenization-makes-data-usable-without-sacrificing-privacy>

👍 🔄 🚩 upvoted 1 times

🗨️ 👤 **Chute5118** 3 years ago

**Selected Answer: D**

Both "Deterministic" and "format preserving" are key-based hashes (and reversible).

It's not clear from the question, but doesn't look like we need it to be reversible.

All of them maintain referential integrity

[https://cloud.google.com/architecture/de-identification-re-identification-pii-using-cloud-dlp#method\\_selection](https://cloud.google.com/architecture/de-identification-re-identification-pii-using-cloud-dlp#method_selection)

👍 🔄 🚩 upvoted 1 times

🗨️ 👤 **cloudprincipal** 3 years, 1 month ago

**Selected Answer: D**

preserve referential integrity and ensure that no re-identification is possible

<https://cloud.google.com/dlp/docs/pseudonymization#supported-methods>

👍 🔄 🚩 upvoted 1 times

🗨️ 👤 **cloudprincipal** 3 years, 1 month ago

forget it, it should be A.

👍 🔄 🚩 upvoted 1 times

🗨️ 👤 **Taliesyn** 3 years, 2 months ago

**Selected Answer: D**

Cryptographic hash (CryptoHashConfig) maintains referential integrity.

"Determinist encryption" is not a transformation method.

<https://cloud.google.com/dlp/docs/transformations-reference>

👍 🔄 🚩 upvoted 2 times

[Load full discussion...](#)



## Platform

> Home

> Examtopics PRO

> All Exams

> Training Courses



