

# Data-Management-Foundations free pdf demo & Data-Management-Foundations training material & Data-Management-Foundations exam prep files

## Demo Overview: Fully Decentralised Authentication Scheme for ICN in Disaster Scenarios (Demonstration on Mobile Terminals)

Jan Seedorf, Bilal Gil, and Dirk Kutscher  
NEC Laboratories Europe  
Heddenberg, Germany  
jan.seedorf@neclab.eu,  
bilal.gil@neclab.eu,  
dirk.kutscher@nedab.eu

Benjamin Schiller and Dirk Kohlweyer  
Technical University of Darmstadt  
Darmstadt, Germany  
schiller@cs.tu-darmstadt.de,  
dirk@kohlweyer.net

### ABSTRACT

Self-certifying names provide the property that any entity in a distributed system can verify the binding between a corresponding public key and the self-certifying name without relying on a trusted third party. However, self-certifying names lack a binding with a corresponding real-world identity. In this demonstration, we present the implementation of a concrete mechanism for using a Web-of-Trust in conjunction with self-certifying names to provide this missing binding. Our prototype runs on Android devices and demonstrates a decentralised message authentication scheme for any kind of content-oriented architecture. In the demonstration, we show how our proposed scheme performs in terms of time needed to assess the trustworthiness of information received – in a fully decentralised scenario: fragmented (mobile) networks. In such a scenario, connectivity to centralised authentication entities and Web-of-Trust key-servers is not available. Our scheme is hence executed solely on end-user terminals itself (which have limited processing capabilities).

### 1. INTRODUCTION

Self-certifying names provide the useful property that any entity in a distributed system can verify the binding between a corresponding public key and the self-certifying name without relying on a trusted third party [3]. This is normally achieved by having the self-certifying name (in some way) on the private key and both of the corresponding public key. Self-certifying names are very useful for addressing the security requirements in *Information Centric Networking (ICN)* architectures. Any source can append a public key and a digital signature (computed with the corresponding private key) to a data item which belongs to a self-certifying name, and any intermediate entity (e.g. an ICN-router/Cache) or any receiving entity (i.e. the holder of an interest for the self-certifying name) can verify the signature with the received public key. The binding between public key and self-certifying name can be verified by anybody, without relying on a trusted third party or a *Public Key Infrastructure (PKI)*. There is thus no need to authenticate the identity of the host that caches an object; the approach does not follow today's *host-centric* security but is inline with ICN's

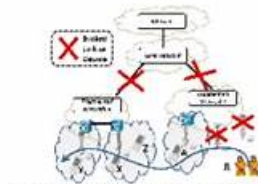


Figure 1: Mobile Network after Disaster with Fragmented Networks [5]

propagated *data-centric* security paradigm. Self-certifying names thus provide a decentralised form of data origin authentication and are very useful in ICN architectures. However, self-certifying names lack a binding with a so-called *Real-World Identity (RWI)* [4]. While the concept enables to verify that whoever signed some data was in possession of the private key associated with the self-certifying name, it does not provide any means to verify what real-world identity corresponds to the public/private key pair, i.e. who actually signed the data [1].

In this demonstration, we present our prototype implementation of a concrete mechanism we have proposed previously [2] for using a Web-of-Trust (WoT) in conjunction with self-certifying names to provide precisely this RWI-binding. We consider a decentralised scenario: fragmented (mobile) networks, when connectivity to centralised authentication entities and WoT key-servers is not available. Our approach enables a particular functionality in this scenario: The assessment of messages from previously unknown publishers. The demonstration will show a prototype running on Android devices.

### 2. DISASTER SCENARIO

Recently, ICN approaches are considered as a solution to enable communication after a disaster took place (e.g. a hurricane, earthquake, or tsunami) [6] [2]. In such a situation, it can be expected that parts of the communication infrastructure have broken down. The (formerly connected) network may be *fragmented* into several islands, e.g. due to failure of certain devices and communication links. Communication resources will be more limited than before the disaster, while at the same time it is important to efficiently distribute key

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided the copies are not made or distributed for profit or commercial advantage, and that copies bear the name and the publication date on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner(s). Copyright is held by the author(s).  
ACM 978-1-4555-2819-2/09/0000...  
ACM 978-1-4555-2819-2/09/0000...  
http://doi.acm.org/10.1145/1608299.1608303

BONUS!!! Download part of Itcertkey Data-Management-Foundations dumps for free: <https://drive.google.com/open?id=1ecSevksBkqGgLDwXEdzdljGWFH8neqq>

Our WGU Data Management – Foundations Exam (Data-Management-Foundations) practice exam can be modified in terms of length of time and number of questions to help you prepare for the WGU real test. We're certain that our Data-Management-Foundations Questions are quite similar to those on Data-Management-Foundations real exam since we regularly update and refine the product based on the latest exam content.

Are you planning to attempt the WGU Data Management – Foundations Exam (Data-Management-Foundations) exam of the Data-Management-Foundations certification? The first hurdle you face while preparing for the WGU Data Management – Foundations Exam (Data-Management-Foundations) exam is not finding the trusted brand of accurate and updated Data-Management-Foundations exam questions. If you don't want to face this issue then you are at the trusted Itcertkey is offering actual and latest WGU Data Management – Foundations Exam (Data-Management-Foundations) Exam Questions that ensure your success in the WGU Data Management – Foundations Exam (Data-Management-Foundations) certification exam on your maiden attempt.

>> Technical Data-Management-Foundations Training <<

**WGU Data-Management-Foundations Paper, Valid Data-Management-Foundations Test Camp**

These WGU Data Management – Foundations Exam (Data-Management-Foundations) certification exam's benefits assist the Data-Management-Foundations exam dumps to achieve their career objectives. To do this you just need to pass the Data-Management-Foundations exam which is quite challenging and demands complete Data-Management-Foundations exam questions preparation. For the quick and complete WGU Data-Management-Foundations PDF Questions preparation you can get help from Itcertkey. The Itcertkey is a leading platform that offers valid, updated, and real Data-Management-Foundations Questions that are particularly designed for quick and complete Data-Management-Foundations exam preparation.

## WGU Data Management – Foundations Exam Sample Questions (Q12-Q17):

### NEW QUESTION # 12

Which phase of entity-relationship modeling refers to the maxima and minima of relationships and attributes?

- **A. Cardinality**
- B. Physical design
- C. Partition
- D. Attribute minimum

**Answer: A**

Explanation:

Cardinality defines the minimum and maximum number of occurrences of one entity in relation to another.

Example Usage in an ER Model:

\* One-to-Many (1:M): A customer can place multiple orders, but each order belongs to only one customer.

Customers (1) --- (M) Orders

\* Cardinality notation:

(1,1) # One-to-One

(0,M) # Zero-to-Many

(1,M) # One-to-Many

Why Other Options Are Incorrect:

\* Option B (Physical design) (Incorrect): Focuses on storage and indexing, not relationships.

\* Option C (Attribute minimum) (Incorrect): No such formal term in database modeling.

\* Option D (Partition) (Incorrect): Refers to dividing tables, not relationship constraints.

Thus, the correct answer is Cardinality, as it defines min/max constraints on relationships.

### NEW QUESTION # 13

Which relationship or association exists between a supertype and its subtype entities?

- A. Associative entity
- **B. IsA relationship**
- C. Weak entity
- D. Strong entity

**Answer: B**

Explanation:

In database modeling, the relationship between a supertype and its subtype is called an IsA relationship.

Example Usage:

\* A Vehicle supertype may have Car and Truck subtypes.

Vehicle

### Car

### Truck

\* In ER diagrams, this is represented as:

Vehicle (Supertype)

|

### Car (Subtype)

### Truck (Subtype)

\* SQL Table Implementation:

sql

CREATE TABLE Vehicle (

VehicleID INT PRIMARY KEY,

```

Make VARCHAR(50),
Model VARCHAR(50)
);
CREATE TABLE Car (
VehicleID INT PRIMARY KEY,
FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID),
EngineType VARCHAR(50)
);
CREATE TABLE Truck (
VehicleID INT PRIMARY KEY,
FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID),
CargoCapacity INT
);

```

\* This structure preserves the IsA relationship between Vehicle (supertype) and Car/Truck (subtypes).

Why Other Options Are Incorrect:

\* Option A (Strong entity) (Incorrect): Strong entities do not rely on a supertype/subtype hierarchy.

\* Option C (Associative entity) (Incorrect): Used to resolve many-to-many relationships, not supertype/subtype relationships.

\* Option D (Weak entity) (Incorrect): Weak entities depend on a strong entity, but supertype-subtype relations use inheritance (not dependency).

Thus, the correct answer is IsA relationship, as it describes the inheritance hierarchy between supertypes and subtypes.

#### NEW QUESTION # 14

Which keyword combines INSERTS, UPDATES, and DELETES operations into a single statement?

- A. DROP
- B. INTO
- C. MERGE
- D. JOIN

**Answer: C**

Explanation:

The MERGE statement, also known as UPSERT, combines INSERT, UPDATE, and DELETE operations into a single statement based on a given condition. It is commonly used in data warehouses and large-scale databases.

Example Usage:

```

sql
MERGE INTO Employees AS Target
USING NewEmployees AS Source
ON Target.ID = Source.ID
WHEN MATCHED THEN
UPDATE SET Target.Salary = Source.Salary
WHEN NOT MATCHED THEN
INSERT (ID, Name, Salary) VALUES (Source.ID, Source.Name, Source.Salary);

```

\* If a match is found, the UPDATE clause modifies the existing record.

\* If no match is found, the INSERT clause adds a new record.

Why Other Options Are Incorrect:

\* Option A (INTO) (Incorrect): Used in INSERT INTO, but does not combine operations.

\* Option B (JOIN) (Incorrect): Used to combine rows from multiple tables, but not for merging data.

\* Option D (DROP) (Incorrect): Deletes database objects like tables, views, and indexes, but does not merge data.

Thus, the correct answer is MERGE, as it combines inserts, updates, and deletes into a single operation.

#### NEW QUESTION # 15

Which primary key values consist of a single field only?

- A. Stable
- B. Simple
- C. Partition

- D. Meaningless

**Answer: B**

Explanation:

A simple primary key consists of only one column that uniquely identifies each row in a table.

Example Usage:

sql

```
CREATE TABLE Students (
  StudentID INT PRIMARY KEY,
  Name VARCHAR(50)
);
```

\* StudentID is a simple primary key because it consists of only one field.

Why Other Options Are Incorrect:

\* Option B (Partition) (Incorrect): Refers to partitioned tables, which divide data for performance reasons but are not related to primary keys.

\* Option C (Stable) (Incorrect): This is not a recognized term in database keys.

\* Option D (Meaningless) (Incorrect): Primary keys are often meaningless (e.g., auto-incremented IDs), but this is not a term used to describe their structure.

Thus, the correct answer is Simple, as a single-field primary key is referred to as a simple primary key.

### NEW QUESTION # 16

Which expression can be used to create a temporary name for a table?

- A. HAVING
- B. UNION
- C. ALIAS
- D. NEW

**Answer: C**

Explanation:

Alias is used in SQL to give a temporary name to a table or column within a query. It makes queries more readable and helps in cases where a table needs to be referenced multiple times (e.g., in a self-join).

Example Usage:

sql

```
SELECT e.Name, d.DepartmentName
FROM Employees AS e
JOIN Departments AS d
ON e.DeptID = d.ID;
```

\* Here, Employees is aliased as e and Departments as d, making the query shorter and clearer.

Why Other Options Are Incorrect:

\* Option A (HAVING) (Incorrect): Used to filter grouped results, not create aliases.

\* Option B (NEW) (Incorrect): Not a valid SQL keyword for aliasing.

\* Option D (UNION) (Incorrect): Combines result sets but does not rename tables.

Thus, the correct answer is ALIAS, which allows for temporary naming of tables or columns.

### NEW QUESTION # 17

.....

Nowadays, the certification has been one of the criteria for many companies to recruit employees. And in order to obtain the Data-Management-Foundations certification, taking the Data-Management-Foundations exam becomes essential. Although everyone hopes to pass the exam, the difficulties in preparing for it should not be overlooked. There are plenty of people who took a lot of energy and time but finally failed to pass. You really need our Data-Management-Foundations practice materials which can work as the pass guarantee.

**Data-Management-Foundations Paper:** [https://www.itcertkey.com/Data-Management-Foundations\\_braindumps.html](https://www.itcertkey.com/Data-Management-Foundations_braindumps.html)

Our Data-Management-Foundations exam test question is the achievement that professional experienced experts worked out

