IoT-Kühlkettenüberwachung

ETS-Supplychain-Projekt

* Phase 2 -

Ein Bild, das Cartoon, Text, Screenshot, Animation enthält.

Automatisch generierte BeschreibungEin Bild, das Screenshot, Text enthält.

Automatisch generierte Beschreibung

Inhalt

[1 IoT-Kühlkettenüberwachung – Projektstufe 2 2](#_Toc164162221)

[1.1 Aufgabenstellung 2](#_Toc164162222)

[1.2 Temperaturüberwachung der Kühlstationen 3](#_Toc164162223)

[1.3 Lieferdatenverschlüsselung 4](#_Toc164162224)

[1.4 Wetterdatenabfrage an den Auslagerorten 5](#_Toc164162225)

[2 Datenmodell 6](#_Toc164162226)

[2.1 Tabellen 6](#_Toc164162227)

[2.2 Gesamtansicht “vcoolchain“ 7](#_Toc164162228)

[3 Daten verschlüsseln und entschlüsseln 8](#_Toc164162229)

[3.1 Symmetrische Verschlüsselung mit AES 8](#_Toc164162230)

[3.2 Passwörter Salzen 8](#_Toc164162231)

[3.3 Padding 9](#_Toc164162232)

[3.4 Die Bibliothek „pycryptodome“ 9](#_Toc164162233)

[3.5 Einen Text verschlüsseln 10](#_Toc164162234)

[3.6 Einen verschlüsselten Text wieder entschlüsseln 11](#_Toc164162235)

[3.7 Datenbankabfrage entschlüsseln 12](#_Toc164162236)

[4 Wetterdaten abfragen 13](#_Toc164162237)

# IoT-Kühlkettenüberwachung – Projektstufe 2

## Aufgabenstellung

Der Hersteller „Food Solution Hildesheim“ möchte die Kühlkettenüberwachung aus Projektstufe 1 durch drei Funktionen ergänzen:

1. Temperaturüberwachung der Kühlstationen
2. Lieferdatenverschlüsselung
3. Wetterdatenabfrage an den Auslagerorten

Bearbeiten Sie die Aufgaben in einer Gruppe, die aus **sechs Personen** besteht.  
Jede Gruppe unterteilt sich in **drei Untergruppen** zu je zwei Personen, die jeweils eines der drei Überwachungskriterien bearbeiten und hierzu einen eigenen Programmteil als „Funktion“ schreiben. Die drei Funktionen für die drei Überwachungskriterien werden in einem Hauptprogramm zusammengeführt.

Planen und dokumentieren Sie ihre Vorgehensweise und die entsprechenden Schnittstellen (z.B. Funktionsnamen, Parameter) schriftlich. Diese Dokumentation soll weiterhin eine grobe Erläuterung der Programmfunktion (max. eine Seite) und den Link auf ein öffentliches Github-Repository (indem der komplette Sourcecode abgespeichert ist) enthalten.

Die Abgabe erfolgt über den Moodle-Kurs.

## Temperaturüberwachung der Kühlstationen

**Scenario:**

Es hat sich herausgestellt, dass trotz der Kühlkettenüberwachung mehrfach minderwertige Ware ausgeliefert wurde, da die Kühltemperaturvorgaben auf dem Transport nicht eingehalten wurden. Zur Qualitätssicherung sollen zukünftig die Temperaturen der Kühlhäuser und der Transportfahrzeuge ausgewertet werden.

**Aufgabe:**

Ergänzen Sie Ihr Programm aus Projektstufe 1 so, dass die Einhaltung der Temperaturgrenzwerte in den Kühlstationen überprüft wird.

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

**Technische Spezifikation:**

* Jede Kühlstation (Güterverteilzentrum und Kühltransporter) schreibt im zeitlichen Abstand   
  von 15 min Temperaturdaten in die Tabelle „tempdata“:
* Die **Kühltemperaturen** dürfen den Bereich zwischen +2°C und +4°C nicht verlassen.

Sie können auch die Sichten „v\_tempdata“ und „v\_tempdata\_crypt“ verwenden!

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

## Lieferdatenverschlüsselung

**Scenario:**

Es hat Beschwerden von Kunden gegeben, dass die Lieferdaten öffentlich einsehbar seien, da sie unverschlüsselt abgespeichert werden. Das hat die Firma „Food Solution Hildesheim“ inzwischen geändert und die Daten verschlüsselt unter „v\_coolchain\_crypt“ abgelegt.

Ein Bild, das Text, Screenshot, Zahl, Software enthält.

Automatisch generierte Beschreibung

**Aufgabe:**

Ändern Sie Ihr Programm aus Projektstufe 1 so ab, dass die verschlüsselten Daten verarbeitet werden können.

**Technische Spezifikation:**

Der Datenbankeinträge sind folgendermaßen verschlüsselt:

* AES-CBC-Verschlüsselung mit Python-Pycryptodome
* Passwort: b'mysecretpassword'
* Initialization-Vector: b'passwort-salzen!'

## Wetterdatenabfrage an den Auslagerorten

**Scenario:**

In der Projektstufe 1 durften die Zeiträume ohne Kühlung, also die Zeit zwischen dem Aus- und Einchecken den Maximalwert von 10 min nicht überschreiten. Für alle gefunden Zeitüberschreitungen soll zukünftig zusätzlich die Temperatur am Übergabeort ausgewiesen werden. So kann die Ware ggf. doch für andere Zwecke weiterverwendet werden.

**Aufgabe:**

Geben Sie für jeden gefunden „***Zeitraum ohne Kühlung“*** aus der Projektstufe 1 zusätzlich die aktuelle Temperatur am Auslagerungsort zur Auslagerungszeit an.

**Technische Spezifikation:**

In der Tabelle „transportstation“ ist die Postleitzahl für jede Transportstation vermerkt.

Da die Transportwagen keinem Ort zugeordnet werden können, ist die Postleitzahl 0 eingetragen.

Ein Bild, das Text, Screenshot, Zahl, Software enthält.

Automatisch generierte Beschreibung

Sie können später auch die Tabelle „transportstation\_crypt“ verwenden!

# Datenmodell

## Tabellen

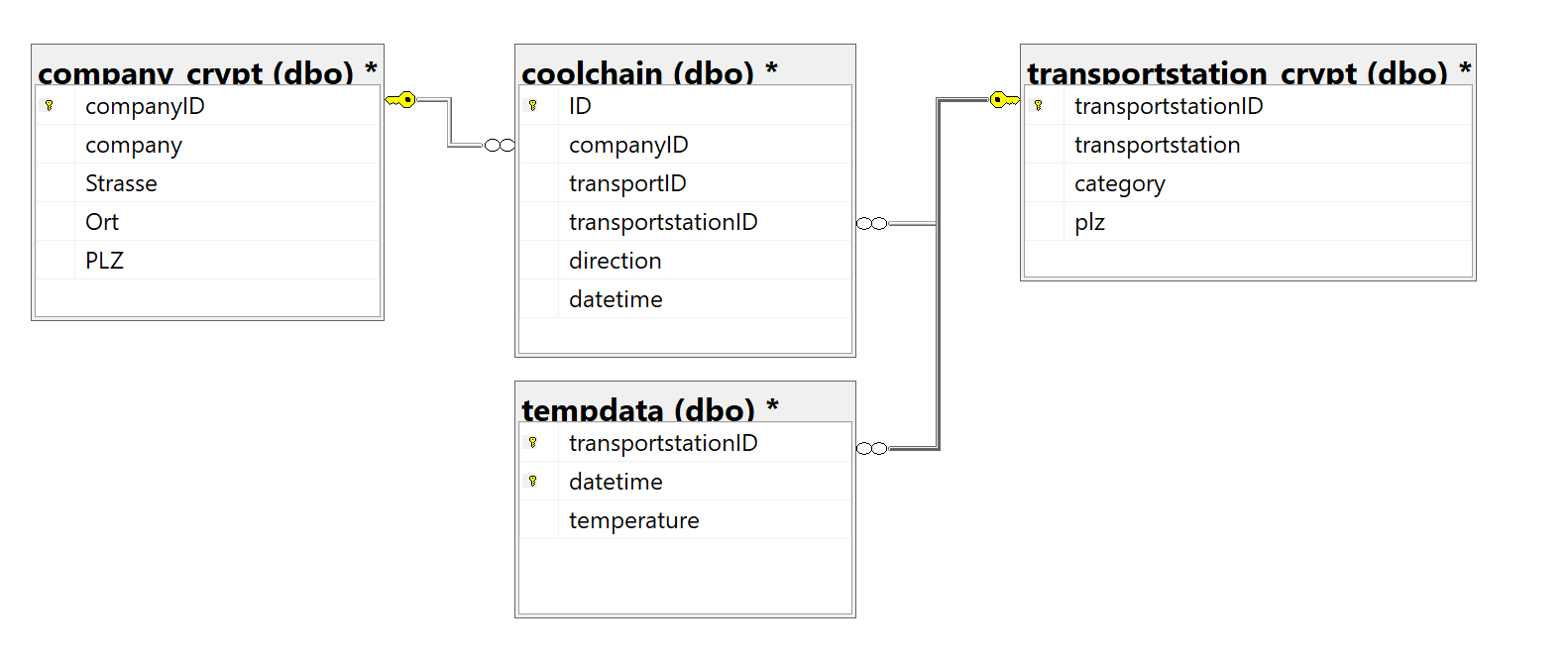
Im Zuge der zweiten Projektstufe wurde die bisherige Tabelle „normalisiert“ und in das folgende Datenmodell überführt. Es besteht aus fünf Tabellen, die über IDs und Fremdschlüssel verbunden sind:

* Tabelle “coolchain“: Transportstationen der Lieferungen
* Tabelle “transportstation\_crypt“: Stammdaten der Kühlhäuser (verschlüsselt)
* Tabelle “company\_crypt“: Stammdaten der Firmen (verschlüsselt)
* Tabelle “tempdata“: Kühlhaus-Temperaturdaten

Im Rahmen der Normalisierung wurden die Daten in **Stammdaten** und **Bewegungsdaten** getrennt.

In den Stammdaten gibt es für jedes Objekt genau einen Eintrag, der mit einer eindeutigen ID als Primärschlüssel versehen wird. Hierzu gehören die Tabellen „company\_crypt“ und „transportstation\_crypt“.

In den Bewegungsdaten „coolchain“ und „tempdata“ werden die IDs der Stammdaten als Fremdschlüssel verwendet.



## Gesamtansicht “vcoolchain“

Die Einzeltabellen können über eine „Sicht“ (View) wieder zusammenfasst werden. Diese Sicht wird über einen SQL-Join-Befehl erzeugt und z. B. unter dem Namen „v\_coolchain\_crypt“ abgespeichert. Aus der Anwenderperspektive verhält sich die „Sicht“ genau wie eine Tabelle. Der Anwender kann in einer „Sicht“ genauso mit einem SELECT-Befehl suchen, wie in einer Tabelle. In der „Sicht“ können jedoch keine Daten verändert werden.

Der SELECT-Befehl für die Sicht „v\_coolchain\_crypt“ sieht folgendermaßen aus.

**SELECT**

company\_crypt.company,

coolchain.transportID,

transportstation\_crypt.transportstation,

transportstation\_crypt.category,

coolchain.direction,

coolchain.**datetime**

**FROM** coolchain

**INNER** **JOIN** company\_crypt

**ON** coolchain.companyID = company\_crypt.companyID

**INNER** **JOIN** transportstation\_crypt

**ON** coolchain.transportstationID = transportstation\_crypt.transportstationID

Diese „Sicht“ ist in der Datenbank unter dem Namen „v\_coolchain\_crypt“ abgespeichert.

Ein Bild, das Text, Schrift, Zahl, Software enthält.

Automatisch generierte Beschreibung

Das Programm aus Phase 1 sollte mit der unverschlüsselten Sicht „v\_coolchain“ statt der Tabelle „coolchain“ weiterhin funktionieren.

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

# Daten verschlüsseln und entschlüsseln

In diesem Beispiel soll ein Text verschlüsselt werden. Hierfür benutzen wir das Verschlüsselungsverfahren Advanced Encryption Standard (AES).

## Symmetrische Verschlüsselung mit AES

AES ist ein symmetrisches Verschlüsselungsverfahren, das einen gemeinsamen Schlüssel für die Verschlüsselung und Entschlüsselung von Daten verwendet. Es funktioniert folgendermaßen:

1. **Schlüsselaustausch**: Bevor Daten gesendet werden, müssen die beteiligten Parteien einen gemeinsamen Schlüssel vereinbaren. Dieser Schlüssel muss sicher übertragen werden, um sicherzustellen, dass keine unbefugte dritte Partei Zugang zu ihm erhält.
2. **Verschlüsselung**: Sobald der Schlüssel ausgetauscht wurde, kann die versendende Partei die Daten mithilfe des AES-Algorithmus und des Schlüssels verschlüsseln. Der AES-Algorithmus teilt die Daten in Blöcke (128 Bit) auf und verschlüsselt jeden Block mit dem Schlüssel.
3. Ein Bild, das Text, Screenshot, Software, Webseite enthält.

   Automatisch generierte Beschreibung**Übertragung**: Die verschlüsselten Daten werden über eine sichere Verbindung an die empfangende Partei gesendet.
4. **Entschlüsselung**: Die empfangende Partei kann die Daten mithilfe desselben AES-Algorithmus und des gleichen Schlüssels, den sie von der versendenden Partei erhalten hat, entschlüsseln.

## Passwörter Salzen

Da der AES-Algorithmus öffentlich bekannt ist, gibt es auch bekannte Angriffsszenarien, um das Passwort zu „knacken“. Wenn man z.B. den Kontext eines verschlüsselten Textes kennt und daher vermutet, dass ein bestimmtes Wort besonders häufig vorkommt (z.B. „BZTG“), kann man den verschlüsselten Text auf Wiederholungen analysieren und durch ausprobieren das Passwort ermitteln.

Um dies zu verhindern „salzt“ man die Passwörter. Für den ersten zu verschlüsselnden Block kombiniert man das Passwort mit einem zusätzlichen Wert, dem „Initialization Vektor“. Für alle weiteren Blöcke nimmt man als Salz das verschlüsselte Ergebnis des letzten Blocks.

So ergibt das gleiche Klartextwort (z.B. „BZTG“) mit dem gleichen Passwort, durch das Hinzufügen verschiedener „Salze“, niemals das gleiche Verschlüsselungsergebnis. Damit ist es unmöglich das Passwort zurückrechnen, indem man den verschlüsselten Text auf Wiederholungen analysiert.

Ein Bild, das Text, Diagramm, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

## Padding

Padding bezieht sich auf den Prozess, bei dem Daten auf eine bestimmte Länge angepasst werden, um sie für eine sichere Verschlüsselung geeignet zu machen.

Im Kontext der AES-Verschlüsselung (Advanced Encryption Standard) müssen die Daten, die verschlüsselt werden sollen, in Blöcke von 128 Bit aufgeteilt werden, und es kann vorkommen, dass ein Block nicht die vollständige Größe hat. In diesem Fall müssen die Daten mit Nullen oder anderen Zeichen gefüllt werden, um die vollständige Blockgröße zu erreichen. Dies wird als „Padding“ bezeichnet.

Es ist wichtig zu beachten, dass das Padding nach der Verschlüsselung entfernt werden muss, um die ursprünglichen Daten wiederherzustellen. Dies nennt man „Unpadding“.

Zur Einführung können Sie folgendes Video ansehen: <https://www.youtube.com/watch?v=AUxHDlC6Rn0>

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Die Bibliothek „pycryptodome“

Pycryptodome ist eine drittanbieter-Bibliothek für Kryptografie in Python. Es ist eine unabhängige Implementierung der Standardbibliothek Python Cryptography Toolkit (PyCrypto) mit aktualisierten Algorithmen und Sicherheitsfunktionen. Pycryptodome bietet eine Vielzahl von Kryptografie-Algorithmen wie AES, DES, RSA und viele andere, die zum Schutz von Daten und zur Authentifizierung von Benutzern verwendet werden können. Es ist ein sehr praktisches Tool für Entwickler, die Kryptografie in ihren Anwendungen einsetzen möchten.

Installieren Sie die Bibliothek „pycryptodome“: py -m pip install pycryptodome

## Einen Text verschlüsseln

# Bibliotheken

from Crypto.Cipher import AES

from Crypto.Util.Padding import pad

# Initialisierung

key = b'mysecretpassword'                # 16 Byte Passwort

iv  = b'passwort-salzen!'                # 16 Byte Initialization Vektor

cipher = AES.new(key, AES.MODE\_CBC, iv)  # Verschlüsselung initialisieren

# Verschlüsselung

plaintext = b'Die ist mein Klartext!'  # Klartext

ciphertext = cipher.encrypt(pad(plaintext,AES.block\_size))  # Text verschlüsseln

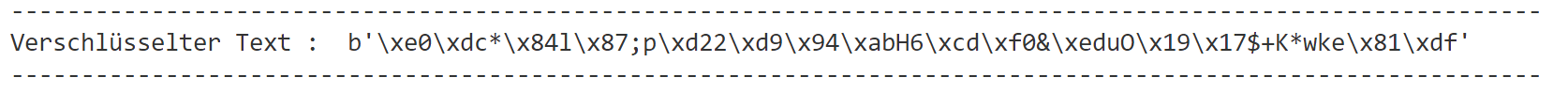
# Ausgabe

print ('-----------------------------------------------------------------------------------')

print ("Verschlüsselter Text : ", ciphertext)

print ('-----------------------------------------------------------------------------------')

Ausgabe:



## Einen verschlüsselten Text wieder entschlüsseln

Als verschlüsselter Text (ciphertext) wird die Ausgabe des vorherigen Programms verwendet.

# Bibliotheken

from Crypto.Cipher import AES

from Crypto.Util.Padding import unpad

# Initialisierung

key = b'mysecretpassword'                # 16 Byte Passwort

iv  = b'passwort-salzen!'                # 16 Byte Initialization Vektor

cipher = AES.new(key, AES.MODE\_CBC, iv)  # Verschlüsselung initialisieren

# Entschlüsselung

ciphertext = b'\xe0\xdc\*\x84l\x87;p\xd22\xd9\x94\xabH6\xcd\xf0&\xeduO\x19\x17$+K\*wke\x81\xdf'

plaintext = unpad(cipher.decrypt(ciphertext), AES.block\_size)  # Text entschlüsseln

# Ausgabe

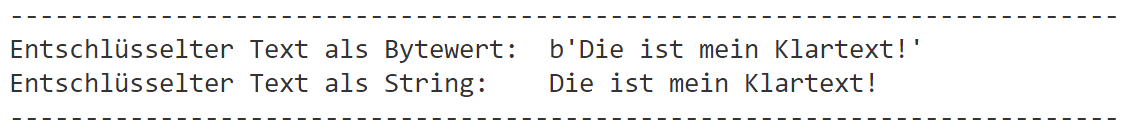
print ('--------------------------------------------------------------------------')

print ("Entschlüsselter Text als Bytewert: ", plaintext)

print ("Entschlüsselter Text als String:   ", plaintext.decode())

print ('--------------------------------------------------------------------------')

Ausgabe:



## Datenbankabfrage entschlüsseln

In dem folgenden Programm werden die Daten der Tabelle „company\_crypt“ entchlüsselt.

import pyodbc

from Crypto.Cipher import AES

from Crypto.Util.Padding import unpad

# Initialisierung

key = b'mysecretpassword'                # 16 Byte Passwort

iv = b'passwort-salzen!'                 # 16 Byte Initialization Vektor

cipher = AES.new(key, AES.MODE\_CBC, iv)  # Verschlüsselung initialisieren

# Entschlüsselungsfunktion

def decrypt\_value(encrypted\_data):

    return unpad(cipher.decrypt(encrypted\_data), AES.block\_size).decode()

# Verbindungsdaten

server = 'sc-db-server.database.windows.net'

database = 'supplychain'

username = 'rse'

password = 'Pa$$w0rd'

# Verbindungsstring

conn\_str = (

    f'DRIVER={{ODBC Driver 17 for SQL Server}};'

    f'SERVER={server};'

    f'DATABASE={database};'

    f'UID={username};'

    f'PWD={password}'

)

# Verbindung herstellen

conn = pyodbc.connect(conn\_str)

cursor = conn.cursor()

# Datensätze auslesen

select\_query = 'SELECT companyID, company, strasse, ort, plz FROM company\_crypt'

cursor.execute(select\_query)

# Für jeden Datensatz die Entschlüsselung durchführen und ausgeben

for row in cursor.fetchall():

   companyID, encrypted\_company, encrypted\_strasse, encrypted\_ort, encrypted\_plz = row

   # Da die Daten als binär gespeichert wurden, sollte hier keine Umwandlung mit str() erfolgen

   decrypted\_company = decrypt\_value(encrypted\_company)

   decrypted\_strasse = decrypt\_value(encrypted\_strasse)

   decrypted\_ort = decrypt\_value(encrypted\_ort)

   decrypted\_plz = decrypt\_value(encrypted\_plz)

   print(f"ID: {companyID}, Company: {decrypted\_company}, Strasse: {decrypted\_strasse}, Ort: {decrypted\_ort}, PLZ: {decrypted\_plz}")

# Verbindung schließen

cursor.close()

conn.close()

# Wetterdaten abfragen

Mit der Wetter-API von „VisualCrossing“ können historische Temperaturdaten an einem spezifischen Datum und einer spezifischen Uhrzeit für einen bestimmten Ort ermittelt werden.

1. **Registrierung**Für die Nutzung des Dienstes benötigen Sie einen (kostenlosen) API-Schlüssel.  
   Registrieren Sie sich auf <https://www.visualcrossing.com/> und kopieren Sie den „key“ aus ihrem „Account“. Fügen Sie den Key später in den Programmcode ein.
2. Ein Bild, das Text, Schrift, Screenshot enthält.

   Automatisch generierte Beschreibung**Bibliotheken**  
   Installieren Sie die Bibliothek „requests“.

import requests

import json

from datetime import datetime

# Beispiel-Nutzung

api\_key = "Hier\_bitte\_Ihren\_API-KEY\_einfügen"

location = "26127,DE"

datetime\_str = "10.07.2023 13:00"  # Zeit auf die nächste volle Stunde gerundet

# Konvertiere das Datum und die Uhrzeit in das erforderliche Format

datetime\_obj = datetime.strptime(datetime\_str, '%d.%m.%Y %H:%M')

timestamp = datetime\_obj.strftime('%Y-%m-%dT%H:%M:%S')

# Visual Crossing Weather API-Endpunkt

url = 'https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/timeline/{location}/{timestamp}'.format(location=location, timestamp=timestamp)

response = requests.get(url, params={'unitGroup': 'metric','key': api\_key,'include': 'hours'})

data = response.json()

# Ausgabe der Temperatur

print("\nTemperatur: ", data["days"][0]["temp"],"\n")

# Ausgabe des gesamten JSON-Objekts

#json\_str = json.dumps(data, indent=4)

#print(json\_str)

Ein Bild, das Schrift, Text, weiß, Grafiken enthält.

Automatisch generierte BeschreibungAusgabe: