DANMARKS STATISTIK

Pseudonymize identifiers for external projects

Amman, February 22, 2024

Forms of pseudonymization in DST

DANMARKS STATISTIK

- There are 2 type of pseudonymization in DST
  - General recode of person-number to person_id in most statistic datasets
    - Handles change of person-number (perople sometimes get a new person-number)
    - Is a number (up to 15 digits)
    - When exchanging back to person-number requires a date.
    - Is being used because it identifies people better, and it also removes person-number from data (person-number is more sensitive)
  - Projectspecific pseudonymization for external projects
    - Pseudonomizes all identifying variables (approx 150 variables)
    - Both for data from DST and for the data that the project may bring in
    - The key is specific for each project

Pseudonymization for external projects

DANMARKS STATISTIK

- A passphrase for each project. Kept in the DDP portal database
- Two algorithms
  - The old one
    - Used for most projects
    - Old. Can be traced back to two punched cards in two different bank vaults in the early 1990'ies.
    - Implemented in SAS, which generates SAS code.
  - The new one
    - Developed in 219/2020 because of the desire to not rely on SAS

Example

DANMARKS STATISTIK

- Different keys for different projects
  - Project 1 key: `Pass1`
  - Project 2 key: `Pass2`
  - Variable value: `0123456789abcd`
  - Pseudonym for variable for project 1: `TKlqHWDufwCd8mRJhvTMRA==`
  - Pseudonym for variable for project 2: `dSeV3K4ryuJj0Mzu0j341w==`

The new algorithm

DANMARKS STATISTIK

- Devloped in IT in cooperation with Research Services
- Needs to not use SAS
  - Not use SAS as a tool and not use SAS-files
    - Need to be able to run on machines, where there are no SAS license
    - There is also a need to be able to handle new dataformats, such at genome data, e.g. PLINK compact)
- Characteristics
  - Must be long lasting (not rely on tools and algorithms that may disappear)
  - Be licensefree, so it can run anywhere
  - Algorithm and implementation must be ready for open sourcing (not rely on security by obscurity).

The new algorithm

DANMARKS STATISTIK

- Characteristics
  - Use accepted crypto-operations for trust
  - Be implementable in different programming languages (e.g python, c#, java)
  - Must accept all types of input (text, binary) and provide text output (text, base64-encoded)
  - Have decent performance.
  - Be applicable in several contexts, e.g standalone for csv-files or as a stored procedure in a database
  - Be reproducible
    - Value pseudonomized with same key always yields same result
  - The algorithm has been assesed by independent auditor (PwC, 2021)

Part of implementation

DANMARKS STATISTIK

```
        _key = _sha_key
    elif AES_KEY_SIZE == 128:
        _key = bytes([_a ^ _b for _a, _b in zip(_sha_key[:AES.block_size], _sha_key[-AES.block_size:])])
    else:
        _key = None
    #
    self.cipher = AES.new(_key, AES.MODE_ECB)
    # Don't leave unneccessary stuff in memory
    password, sha_key, _sha_key, _key, = None, None, None, None

def encrypt(self, s: bytes):
    """The encrypted bitstream is returned base64 encoded"""
    return b64encode(self.cipher.encrypt(pad(s, AES.block_size)))

def decrypt(self, s: bytes):
    """The plaintext is expected to be base64 encoded"""
    return unpad(self.cipher.decrypt(b64decode(s)), AES.block_size)
```

output

DANMARKS STATISTIK

```
0000000000;ffJbtH2t91rMaocb7fcUdyuw==
0000000001;ffjGro1r/QD8utL19CWNkByw==
0000000002;ff74EuplMs6z3itxV55MOrxA==
0000000003;ffWBzj0y3c9CetQVKwLtTR1A==
0000000004;ffd6nEzIxEfNeYeivA4yKmLA==
0000000005;ffCnIoWnvBEQ1NpRCvd+Xfyg==
0000000006;ff43o9IWudREKVCDFY58CZsA==
0000000007;ffP+G1i+l6yuXwgoanKqc6pw==
0000000008;ffmpUkVZn8EDblLz10UK8t6Q==
0000000009;ffiSE4NKv7lxGfCf4b4DNOaQ==
```