**Linkja-Hashing**

Linkja is a java tool to obfuscate PHI. It creates several permutations & combinations from the patient identifiers, hashes the values with salt & crypto algorithm provided by the salt generator, encrypts the hashed records with the public key provided by the aggregator, and packages the output as a .enc file. The tool works in-memory to accomplish above tasks, thus it does not store the real salt used in hashing or pre-encrypted values of data.

To begin, create a project folder to store all project specific files:

* linkjacrypto.dll: Salt-crypto generating agency sends this file. It contains secret algorithm to generate salt for hashing in memory
* saltFile: Salt-crypto generating agency sends this file. It contains seed to be used by the algorithm to generate project salt and site salt in memory
* encryptionKey: Aggregator sends this file. It contains RSA 2048-bit public key to be used in encrypting the hashes
* jar: Latest version should downloaded from the github release page (<https://github.com/linkja/linkja-hashing/releases>)
* project-data: This file contains patient identifiers to be used in hashing (patient id, name, dob, and optional ssn)

**Command:**

On command line (cmd), cd (i.e., change directory) to your project directory. And provide below arguments to begin hashing

Key arguments

-jar <arg> Name of jar file

--encryptionKey <arg> File name to public key sent by aggregator

--saltFile <arg> File name to salt file sent by salt-crypto

--patientFile <arg> File containing patient identifiers (.csv or .txt)

--privateDate <arg> Any random date in format: MM/DD/YYYY

--delimiter <arg> If patient file is .txt, then delimiter used

**Example:**

java -Djava.library.path=. -jar Hashing-0.9-jar-with-dependencies.jar --encryptionKey public-agg.key --saltFile project\_x12\_006\_20200509.txt --patientFile project-data.csv --privateDate 01/01/2018

Note: Please update the filenames in above command with your project files and latest jar file.

**Outputs:** There are 3 outputs generated by Linkja

1. encrypted file: This file has an extension of “.enc”. It contains hashed identifiers and should be shared with the aggregator for matching

2. crosswalk: This file contains crosswalk between site’s local patient ids and hashed patient id. This crosswalk can be used to link back to site’s data

3. invalid data file: This file contains records that linkja could not process due to errors in data (e.g., missing first name). To process these records, fix the error and re-run linkja

For testing, please download:

1. <https://github.com/linkja/linkja-test-data/tree/master/hashing>
2. <https://github.com/linkja/linkja-test-data/tree/master/crypto/secret>

**Appendix**: Patient data

The patient data should be csv or text delimited file (headers required)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field Name | Data Type | Description |
| 1 | Patient ID | string or number | Unique patient identifier\* |
| 2 | First Name | string | Patient first name |
| 3 | Last Name | string | Patient last name |
| 4 | DOB | string | Date of birth\*\*\*\* |
| 5\*\* | SSN | string | Social Security Number\*\*\* |

\*Each patient record should have a unique ID

\*\* SSN is optional, if the SSN is missing, hashes that require SSN (e.g. hash1 - fnamelnamedobSSN) will remain blank.

\*\*\* The application can handle last 4 digits of SSN as well as full Social Security Number (only last 4 numbers get used in hashing)

\*\*\*\*The application can handle several date formats YYYY-MM-DD (e.g. 1960-12-31), YYYYMMDD (e.g. 19601231), MM/DD/YYYY (e.g. 31/12/1960). Select 1 type of format for all records.

**Appendix**: Output Hashed Encrypted file (.enc extension)

This is the only file that should be shared with the aggregator. Linkja in memory, hashes the record, encrypts the hashed records, and packages the output as a .enc file. Only the aggregator with private key can open this file. The file contains below values (the values will not be visible to the site as only aggregator can open this file):

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field Name | Data Type | Description |
| 1 | Site ID | String | Site ID |
| 2 | Project ID | String | Project ID |
| 3 | PIDHASH | String | Patient ID + Site ID + Date Offset (Private Date and DOB) |
| 4 | hash1 | String | First Name + Last Name + DOB + L4 SSN |
| 5 | hash2 | String | Last Name + First Name + DOB + L4 SSN |
| 6 | hash3 | String | First Name + Last Name + DOB |
| 7 | hash4 | String | Last Name + First Name + DOB |
| 8 | hash5 | String | First Name + Last Name + Transposed DOB + L4 SSN |
| 9 | hash6 | String | First Name + Last Name + Transposed DOB |
| 10 | hash7 | String | 3 Initials First Name + Last Name + L4 SSN |
| 11 | hash8 | String | 3 Initials First Name + Last Name + DOB |
| 12 | hash9 | String | First Name + Last Name + DOB +1D + L4 SSN |
| 13 | hash10 | String | First Name + Last Name + DOB +1Y + L4 SSN |

DOB=date of birth (YYYY-MM-DD)

Transposed DOB = Month and Date Transposed in date of birth (YYY-DD-MM)

1D = 1 day offset in date of birth

1Y = 1 year offset in date of birth

L4 SSN = Last 4 Social Security Numbers

Fields 3 – 13 are SHA256[[1]](#footnote-1) hashes (64 hexadecimal characters)

1. https://en.wikipedia.org/wiki/SHA-2 [↑](#footnote-ref-1)