Share Analysis. Not Data.

PRANA-DATA

DA.3 Proof of principle of an approach based on bringing the algorithms to the data – patient setting

Project	PRANA-DATA
Project leader	Wessel Kraaij (TNO)
Work package	PoP Patient Setting
Deliverable number	DA.3
Authors	Thymen Wabeke (TNO)
Reviewers	Wessel Kraaij
Date	March 29, 2017
Version	1
Access Rights	Public
Status	Final

PRANA-DATA Partners:

Portavita, TNO, Radboud Universiteit Nijmegen, Maastricht UMC+, UMCG

COMMIT/ COMMIT is a public-private research community solving grand challenges in information and communication science shaping tomorrow's society

Summary

This document describes the development of a proof-of-principle that predicts growth evolvement of infants without the need to share sensitive user data in plain-text. All analyses are performed in the homomorphic domain such that (parents and doctors of) infants gather novel insights while third parties don’t learn anything from the sensitive user data.

Contents

Summary 1
1 Introduction 3
1.1 Predict growth using curve matching 3
1.2 Privacy-enhanced analyses using homomorphic encrypting 4
2 Secure Patient Matcher (SPM) 5
2.1 Prediction pipeline 6
3 References 7

1 Introduction

Comparing and enriching user profiles using data of different parties can reveal novel and interesting insights. However, this usually comes with an undesirable side effect: the party that combines and analyses profiles obtains access to sensitive data. A proof-of-principle was developed in the PRANA-DATA project that allows a third party to analyze and combine homomorphically encrypted user profiles. By encrypting user profiles, sensitive data remains unknown to the third party while new insights can still be gathered.

The proof-of-principle was developed for the use case of predicting growth development of young infants. It can be seen as a more secure and privacy enhanced implementation of the "curve matching" method proposed by van Buuren (2014). The present document motivates and describes this proof-of-principle called Secure Patient Matcher (SPM). The first sections briefly explain curve matching and homomorphic encryption. Section 2 explains the architecture and analysis pipeline of the Secure Patient Matcher.

1.1 Predict growth using curve matching

Imagine Alice, a three months old baby. We would like to predict Alice's growth development and estimate whether growth issues such as stunting and obesity are likely to occur when she is a toddler. These growth predictions can be made using curve matching (van Buuren, 2014). This technique exploits historical growth data and finds existing children in the historical database that are similar to the current child (e.g., Alice). The growth patterns of the matched children suggest how the current child might evolve in the future (see Figure 1). We use the SMOCC dataset containing historical growth data of about 2000 children (Herngreen et al, 1992). Homomorphic encryption was used to predict in a privacy enhanced manner that does not require sharing plain data.

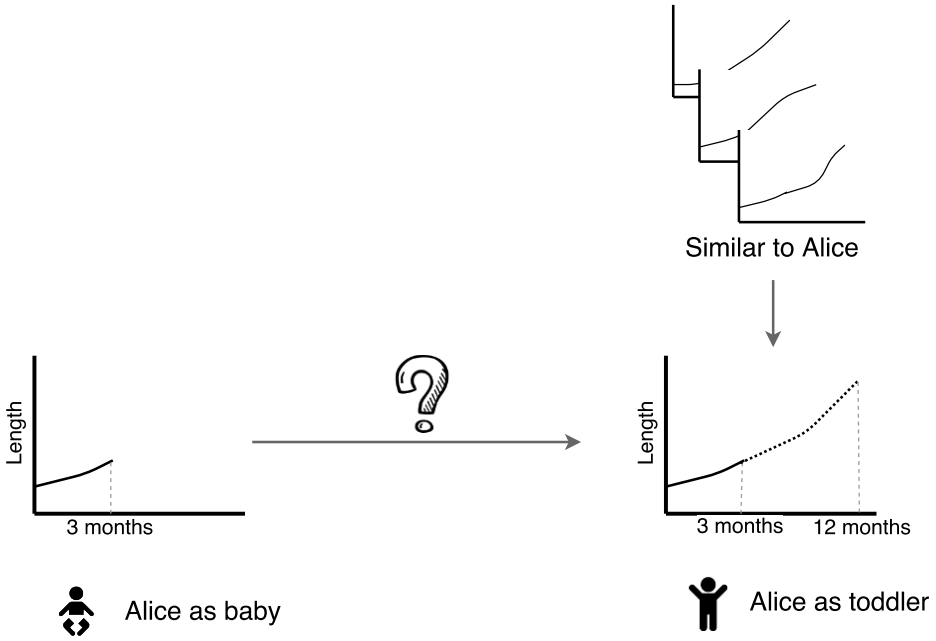

Figure 1: Curve matching is used to predict growth evolvement of young infants using historical data of similar children. Homomorphic encryption was applied to predict in a privacy enhanced manner.

1.2 Privacy-enhanced analyses using homomorphic encrypting

User data is usually analyzed in the plain-text domain. By doing so, parties that process data get to know sensitive information about users. Data can be encrypted to enhance privacy. Encryption turns plain-text data into unreadable chipertext. Encrypted data can usually only be read, analyzed or altered when one has access to a proper decryption key. This characteristic makes traditional encryption unsuited for privacy-enhanced data analyses.

Using homomorphic cryptosystems, however, operations can be performed on ciphertexts such that decrypting the result of the operation is the same as the result of some operation performed on the plaintexts. Most homomorphic cryptosystems support either additive or multiplicative operations and are called partly-homomorphic. ElGamal (multiplicative) and Paillier (additive) are popular partly-homomorphic cryptosystems. Fully-homomorphic cryptosystems support both multiplicative and additive operations in the encrypted domain. Hence, these systems can (in theory) support all possible data analysis.

In practice both partly and fully homomorphic cryptosystems have drawbacks. For instance, an increase of computation time and allocated memory. Furthermore, only a few stable homomorphic implementations of popular analyses like regression, random forests and support vector machines exists.

A proof-of-concept for the curve matching analyses was developed in this project using Paillier's partly-homomorphic cryptosystem[1]. Sensitive user data was encrypted in the user domain before sending it to the party performing the analyses. In this manner, the analysis party could compute with the sensitive data without learning anything from it. Figure 2 displays this procedure.

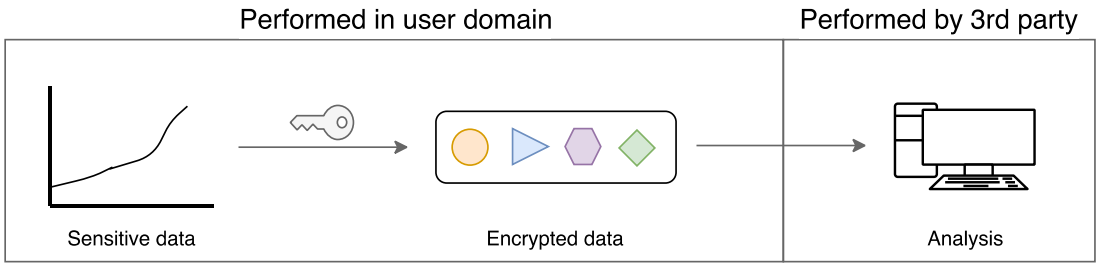

Figure 2: Sensitive data is homomorphically encrypted in the user domain. Analyses are performed using this encrypted data. This procedure enhances privacy because third parties are unable to read sensitive user data.

[1] https://en.wikipedia.org/wiki/Paillier_cryptosystem

2 Secure Patient Matcher (SPM)

The Secure Patient Matcher predicts growth patterns of young children using homomorphically encrypted data records of peers. We use a semi-honest security model. This means that all components/parties are curious: they want to know as much as possible. However, they are honest and do follow the protocol. A simplified architecture of the Secure Patient Matcher proof-of-principle is displayed in Figure 3. Section 2.1 describes the steps included in this architecture.

The proof-of-principle was developed in Python. The components were packaged Docker images, communicate with each other using web-API's and can operate in different environments. For example, the analysis and decryption server might be run by different parties.

All homomorphic operations were performed using the "python-paillier" library[2]. This library was developed by the Australian research institute CSIRO and implements the Paillier cryptosystem. The library integrates with popular data analytics tools like Numpy. Although not all operations were supported by this library, the integration with other tools is an advantage when applying homomorphic encryption in a real-world scenario.

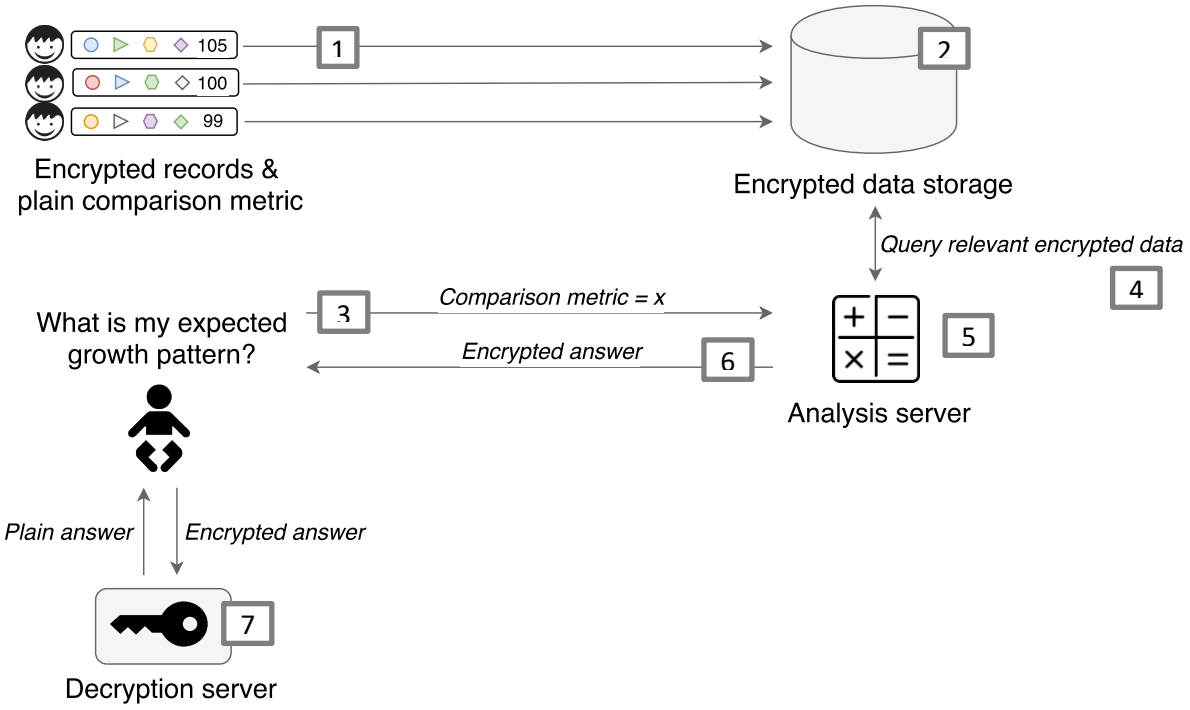

Figure 3: Simplified architecture of the Secure Patient Matcher proof-of-principle.

[2] https://pypi.python.org/pypi/phe/

2.1 Prediction pipeline

Predicting the growth pattern of a young baby using homomorphic encryption is performed using these steps:

1. Bob and other users have their own historical growth data and a comparison metric, which is the length measured at the age of twelve months. The growth data is homomorphically encrypted in the user domain. Encryption is performed using a single public key and makes the data unreadable for all other parties. The public key cannot be used to decrypt any data. The comparison metric is not encrypted as it doesn't represent sensitive information.
2. The encrypted growth data and comparison metrics are stored in a data storage.
3. Parents of young children can ask questions to the analysis server now. Alice('s father), for example, would like to know her expected growth pattern because this prediction indicates whether growth issues are likely to occur. Alice also sends her comparison metric when asking the question. There are models available to calculate a comparison metric based on whether a child is a twin member, the mother smoked during pregnancy, etc.
4. The analysis server queries relevant encrypted data from the data storage. This concerns data of the five children having a comparison metric most similar to that of Alice.
5. The analysis server aggregates the data of these similar users in the homomorphically encrypted domain. The input data and obtained answer are both encrypted such that the server doesn't learn any sensitive information about the children. The server does learn the plain comparison metrics. However, these values are not linked to known individuals and represent the (expected) length at the age of twelve months. Eventually, the analysis server thus learns something which is already publically known, namely the distribution of lengths.
6. The analysis server sends the encrypted answer back to Alice.
7. Alice forwards the encrypted answer to the decryption server. This is the only party having access to the private key which is required for decryption. Alice can add a random number before forwarding the answer. The decryption server then doesn't learn the actual answer but the sum of the answer and a random number. After decryption, Alice subtracts the random number from the decrypted data to obtain the actual answer. Alice learns her expected growth pattern now.

3 References

van Buuren, S. (2014). Curve matching: A data-driven technique to improve individual prediction of childhood growth. *Annals of Nutrition and Metabolism*, *65*(2-3), 227-233.

Herngreen, W. P., Reerink, J. D., van Noord-Zaadstra, B. M., Verloover-Vanhorick, S. P., & Ruys, J. H. (1992). SMOCC: Design of a representative cohort-study of live-born infants in the Netherlands. *The European Journal of Public Health*, *2*(2), 117-122.