CONFERENCE OF EUROPEAN STATISTICIANS

Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality Session 4

29-31 October 2019, the Hague, the Netherlands 18 October 2019

Privacy Preserving Set Intersection

Giuseppe Brun and Diana Nicoletti (Bank of Italy), Monica Scannapiecoy and DiegoZardetto (Italian National Statistical Office, Italy)

Giuseppe.Bruno@bancaditalia.it

Abstract and Paper

Modern societies are increasingly dependent on (and sometimes afraid of) massive amounts and availability of digital information. There are numerous scenarios where sensitive data must be – even if reluctantly or suspiciously - shared between entities (or institutions) without mutual trust. National privacy protection laws, however, for- bid the processing of non-anonymized records between institutions and even within the same institution, thus making it difficult to carry out statistical studies. The goal of this paper is to explore viable techniques to carry out data linkage among institutions while preserving a desirable level of anonymity. Taking advantage of cryptographic techniques introduced since the 90s, we provide some examples of linkage of anonymized files, in an attempt to overcome the current privacy constraints. Administrative records including socio-economic and financial information, both for households and firms, have huge potential for statistical studies. In Italy, the Fiscal Code is a widely used identifier for individuals and firms, and this would theoretically provide wide scope for data matching, provided that the legislation on such matters was respected. In this paper, we propose some modifications on a protocol, based on hashing and cryptographic functions, to strongly distinguish among identifying and not-identifying data and carrying out the objective of performing privacy preserving analytics. The examined scenarios offer a strict supervision over who is in possession of which information. This allows us to prevent unauthorized linkage of data and to protect individuals anonymity. Here our goal is to explore viable techniques to carry out data linkage among institutions while preserving a desirable level of anonymity. We present some examples of empirical applications with the use of synthetic data.

Keywords

Data linkage, hash functions, encryption, record re-identification

PRIVACY PRESERVING SET INTERSECTION [1]

Giuseppe Bruno[*], D. Nicoletti[*], M Scannapieco[**] and D. Zardetto[**]

(*) Bank of Italy Economics and statistics department.

(**) ISTAT Italian National Statistical Office.

Abstract: Modern societies are increasingly dependent on, and sometimes afraid of, huge amounts of information. There are numerous scenarios where sensitive data must — even if reluctantly or suspiciously — be shared between entities (or institutions) without mutual trust. National privacy protection laws, however, forbid the processing of non-anonymized records between institutions and even within the same institution, thus making it difficult to carry out statistical studies. The goal of this paper is to explore viable techniques to carry out data linkage among institutions while preserving a desirable level of anonymity. Taking advantage of cryptographic techniques introduced since the 90s, we provide some examples of linkage of anonymized files, in an attempt to overcome the current privacy constraints.

Keywords: Data linkage, hash functions, encryption, record re-identification.

[1] The views expressed in the paper are the authors' only and do not imply those of their institutions

Table of contents

1. Introduction

Modern societies are increasingly dependent on, and sometimes afraid of, massive amounts and availability of digital information. There are numerous scenarios where sensitive data must — even if reluctantly or suspiciously — be shared between entities (or institutions) without mutual trust.
Administrative records including socio-economic and financial information, both for households and firms, have huge potential for statistical studies. In Italy, the Fiscal Code is a widely used identifier for individuals and firms, and this would theoretically provide wide margin for data matching, provided that the legislation on such matters was respected.
The law, however, forbids the processing of non-anonymized records within and between institutions, thus making it difficult to carry out statistical studies.
We would like to benefit from computer science, specifically cryptographic techniques introduced since the 90s, to provide safe linkage of anonymized files, in order to overcome the current constraints of such procedures.
In this paper, we propose some modifications on a protocol, based on hashing and cryptographic functions, to strongly distinguish among identifying and not-identifying data and carrying out the objective of performing privacy preserving analytics.
The examined scenarios offer a strict supervision over who is in possession of which information. This allows us to prevent unauthorized linkage of data and to protect individuals anonymity.
Here our goal is to explore viable techniques to carry out data linkage among institutions while preserving a desirable level of anonymity. We present some examples of empirical applications with the use of synthetic data.
Our view is that the main obstacle to setting up such a system is not technical, but rather organizational in that it is based on the concept of a third party as trusted Authority.
Although prior work has yielded a number of effective and elegant Private Set Intersection (PSI) techniques, the quest for efficiency is still underway. Here we propose some PSI variations of a well-known algorithm and security improvements that scale well up to a billion records.
The results achieved so far seem very promising therefore we deem of paramount importance the extension of our work along the following two lines:

1) a deeper experimentation with real data, different tools and protocols,
2) the cooperation of the Data Protection Authorities to play a specific role in the protocols;

The rest of the paper is organized in the following way: in the next paragraph we explain the need for Private Data Sharing Protocols and, later, we describe four reference scenarios that can possibly take place when a private information sharing need occurs (section 3). Then, section 4 provides a broad illustration of the two practical exercises that were implemented as empirical applications. Finally, section 5 draws the main conclusion of this paper.

2. Why Private Data Sharing Protocols

Privacy by Design is an approach to information system engineering that takes privacy into account throughout the whole engineering process. From a legislative perspective, the

concept of Privacy by Design has been formally introduced in the EU Regulation 2016/679, where Article 25 is about "Data protection by design and by default".
The Italian National Institute of Statistics (Istat) and the Bank of Italy started a research collaboration with the following two goals:

1) Trying to figure out what concretely "privacy by design" can mean in a multi-organizational information sharing context.
2) Setting up some experiments to validate the feasibility, the performance and the open issues on adopted methods.

With the increasing presence of anytime-anywhere data, there are many realistic scenarios where sharing data among two parties could be beneficial for both of them. Even with the assumption of honest behavior from the two parties, there is always the need to devise robust protocols providing high security standards along with a limited computational power.

3. Scenarios for private data sharing

We envision four scenarios that can support the "generic" information sharing need, namely:

i) private set intersection (PSI);
ii) private set intersection with enrichment (PSI-E);
iii) private set intersection with analytics (PSI-A);
iv) private data mining.

3.1 Private Set Intersection (PSI)

Here two further specific scenarios can be identified, namely Exact and Approximate PSI.

Exact PSI Definition: Let P1 and P2 be parties owning (large) private databases D1 and D2. The parties wish to apply an exact join to D1 and D2 without revealing any unnecessary information about their individual databases. That is, ideally, the only information learned by P1 about D2 and by P1 about D1 is $D1 \cap D2$.

Approximate PSI Definition: Let P1 and P2 be parties owning (large) private databases D1 and D2. The parties wish to apply an approximate join to D1 and D2 without revealing any unnecessary information about their individual databases. That is, ideally, the only information learned by P1 about D2 and by P2 about D1 is $D1 \cap D2$.
An example of an exact PSI algorithm can be found in [1], while an example of an approximate PSI algorithm can be found in [2].
Differently from exact joins, approximate joins are an indirect, somewhat complicated process, as they require the computation of distance functions among records, the values of which have to be kept private[2].

[2] As an example, computing a distance dist(d1,d2), where d1 and d2 are two data items owned respectively by sources P1 and P2, requires both values to be available at the same time to one party. However, under privacy constraints, such simple condition cannot be met in that P1 cannot see d2 and P2 cannot see d1. Even if a third neutral party is introduced, it cannot compute neither the distance between the plain values of d1 and d2, nor the distance between encrypted values, because encryption functions do not generally

3.2 Private Set Intersection with Enrichment (PSI-E)

PSI-E Definition Let P1 and P2 be parties owning (large) private databases D1 and D2. The parties wish to apply an exact or approximate join to D1 and D2 without revealing any unnecessary information about their individual databases. After that, they wish to enrich joined records with variables by both parties. At the end of the process P1 will learn additional P2 variables on D1∩ D2 and P2 will learn additional P1 variables on the same intersection (see Figure 1).

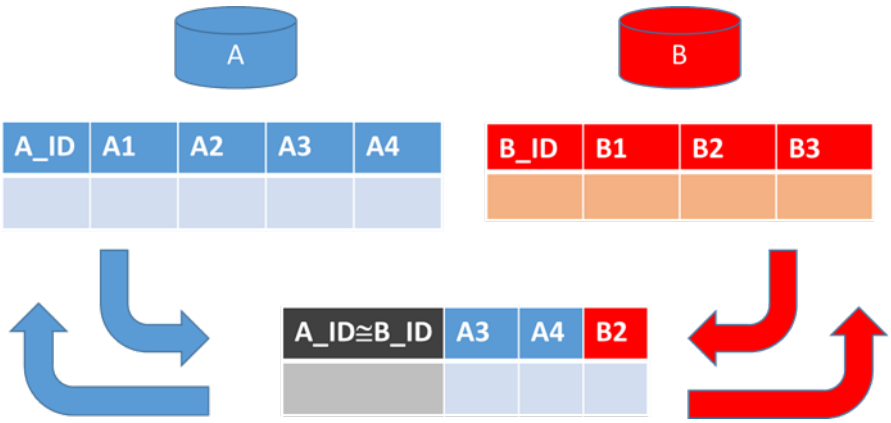

FIGURE 1: PRIVATE SET INTERSECTION WITH ENRICHMENT SCENARIO

3.3 Private Set Intersection with Analytics (PSI-A)

PSI-A Definition. The parties wish to perform a statistical analysis on the intersection of their databases in a private fashion. To identify the records belonging to the intersection, they agree to apply an Exact PSI. At the end of the process, the only information learned by the parties (beyond the keys of the records belonging to the intersection) is the result of the statistical analysis.

3.4 Private Data Mining

Private data mining. "Let P1 and P2 be parties owning (large) private databases D1 and D2. The parties wish to apply an analytics function to the joint database D1∪D2 without revealing any unnecessary information about their individual databases. At the end of the process, the only information learned by P1 about D2 is that which can be learned from the output of the analytics function, and vice versa " [3].

4. Case Study n. 1

4.1 General considerations

preserve similarity distances. In order to overcome such a problem ad-hoc protocols need to be proposed that may require complex data processing.

For this empirical application we have considered the interaction between two institutions that strictly adhere to the proposed protocol. This kind of behavior, in the literature, is usually defined as Honest but Curious (HbC). Within this framework, the institutions will not try to make queries purposefully designed to gain other attributes on particular individuals, for example by sending just one or multiple identical keys to receive the attributes owned by the server partner. A possible way to reduce the chance of re-identification would be the check that the number of different key-values is greater than a given threshold.

4.2 Description of the use case

This case study deals with an application of PSI-E described in chapter 0. We consider two organizations having two basic health information datasets. The record structure of the first dataset is the following:

1) Individual name; this field is a key in the whole table.
2) Weight, a positive continuous variable
3) Smoker, a binary variable (1=smoker, 0=not-smoker)

The second dataset has the following structure:

1) Individual name; this field is a key in the whole table.
2) Height, a positive continuous variable,
3) Blood_pressure, a positive continuous variable.

The owners of the two datasets wish to join them for statistical purposes without making the names available to the other party. In this empirical application we assume each party as honest but curious. This categorization imply that the institutions will not try to cheat the other party not abiding the protocol rules. This assumption is tantamount to say that institutions will not exchange dataset whose attributes are distributed in such a way to allow straight record re-identification.
The interaction between the two institutions yields a joined dataset which retains the common elements without any reference to the individual's name. The following picture describes the employed testing framework.

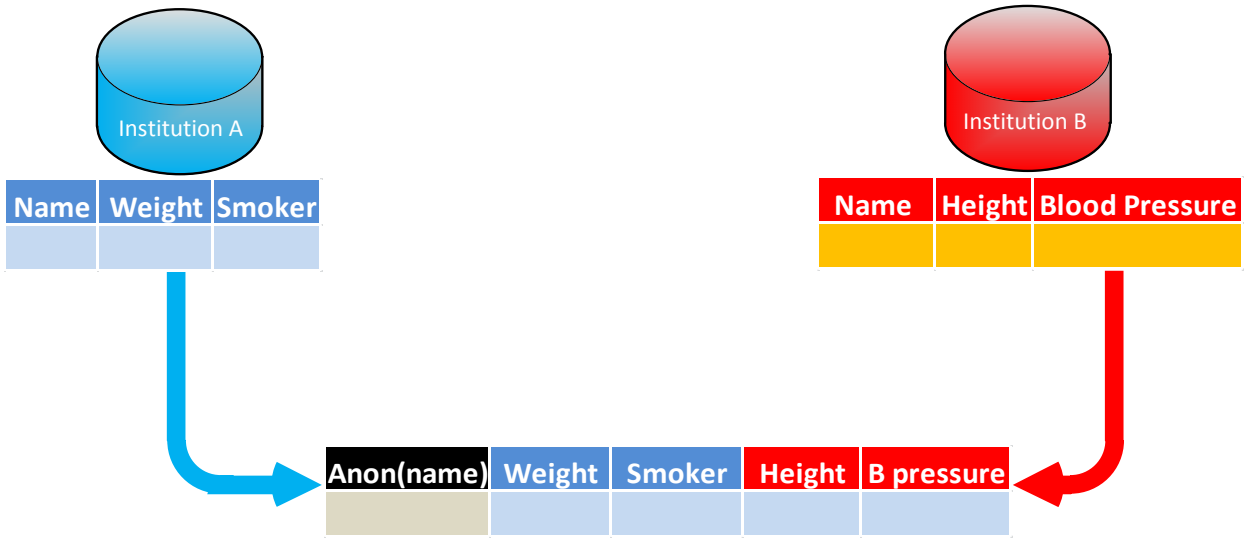

FIGURE 2: CASE 1 EXAMPLE

As can be seen from Figure 2 the shared dataset includes the variables provided by the two institutions but the identification key has been anonymized by a sequence of hashing and encryption steps which prevents re-identification from either parties.

4.3 Description of the Protocol

The protocol employed for the implementation of this case study is based on De Cristofaro-Tsudik (2010) (DCT). The Private Set Intersection (PSI) is a cryptographic protocol entailing two players, in our setup two institutions, which are called client and server.
In this empirical application we have considered a mutual version of the protocol where the roles of client and servers are exchangeable between the two institutions. Within this framework the outcome is available to both the players.
We have employed a modified version of the "Blind RSA-based PSI protocol" proposed in DCT. The original protocol has been improved along the following lines:

1) A proxy server has been put in front of the web server providing the data (for IT security purpose);
2) A code parallelization has been carried out for performance improving;
3) The communication between the client and the proxy server is encrypted through a digital certificate;

The protocol implementation has been carried out in Python starting from a version by Constantinos Patsakis available at: *https://github.com/kpatsakis/PSI_De_Christofaro* .
The protocol is based on the RSA[3] encryption algorithms and the SHA256[4] (256 bit Secure Hash Algorithm). RSA security is rooted on the key-length while resilience of hashing function comes from the length of the output digest. In our experiments we have checked the protocol performances by varying RSA key-length and the output digest length.

5. Case Study n. 2

5.1 Description of the use case

This case study has been developed with reference to the PSI-A scenario introduced in Section 2.3. The two involved parties own databases D1 and D2 respectively. D1 and D2 have a common key, which can be exploited to perform an Exact PSI. The parties wish to enrich their information assets by learning the results of a statistical analysis[5] applied to the *intersection* of their databases. This goal must be reached in compliance with *all* the following requirements:

- only the strictly necessary data are transmitted;
- only encrypted data are transmitted;
- secure data transmission protocols are used;
- the intersection of private databases is obtained by an Exact PSI;

[3] RSA stands for Rivest, Shamir and Adleman, the three researchers who proposed the algorithm in 1978.

[4] See for example http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf

[5] In this preliminary phase, we have considered a very simple statistical analysis, namely the computation of absolute frequency distributions. Therefore, the parties will be able to learn only counts of units classified by a set of categorical variables.

- the parties learn *only* the results of the required statistical analysis (beyond the keys of the records belonging to the intersection);

To be more specific, let us suppose that D1 database has variables 'tax code' (common key), 'number of children' and 'age class', while the D2 database has variables 'tax code' (common key), 'income class', 'mortgage payment class' and the binary variable 'solvent/insolvent borrower'. The objective for both parties is to privately query the intersection of their databases and retrieve counts with respect to a given set of grouping variables. For instance, the first institution could be interested in learning how many insolvent borrowers have more than 2 children, e.g. in the context of actuarial risk modelling. The second institution, in turn, might want to know how many elderly people (aged 65 or more) belong to the lowest income class, e.g. in the context of poverty analysis. Note that in both these examples, each party takes advantage of variables which it does *not* own, but which are *privately provided* by the other party.

5.2 Description of the use case

To implement the use case described above, we assume to be in an honest-but-curious (HbC) context, so that all the involved parties will respect all the rules defined in the protocol.
The protocol requires, in addition to the parties who want to privately share their data, a third party named *Linker*. The role of the Linker is to:

- receive and store encrypted data from both parties;
- receive queries from the parties, process the encrypted data and return to the parties the results of the queries they submitted.

Considering its role, the Linker should be a *super partes* organization trusted by all the Institutions participating to the protocol. In this experimental phase we might assume this role is alternatively played by one of the institutions. From a theoretic perspective its role could be played by a national or super national Date Protection Authority.
The experimental protocol implemented requires the following four phases (the purpose of each phase is sketched in parentheses):

1) *Preliminary phase* (agreement between the parties on base protocol parameters)
2) *Exact PSI* (private intersection of common database keys)
3) *Loading* (transmission of encrypted data to the Linker)
4) *Query* (submission of queries to the Linker and transmission of results)

In the *Preliminary phase*, the parties agree on: (a) the name of the common database variable to be used for the Exact PSI, (b) the names of the database variables they want to share, (c) the symmetric cryptographic key that both parties will use to encrypt their private data (note that this symmetric key is transmitted through the RSA protocol).
In the *Exact PSI* phase, the PSI protocol of De Cristofaro [1] is applied, with the two parties playing alternately the client and the server roles. At the end of the phase, both parties have learned the keys of the records belonging to the intersection of their private databases.
In the *Loading* phase, the parties upload to the Linker the encrypted version of the data they want to share. More precisely, for each record of the intersection determined in the

Exact PSI phase (and only for them), the parties provide the encrypted values of their private variables.
In the *query* phase, first, the parties asynchronously submit a query to the Linker; then, the Linker computes the query results by processing the encrypted data; finally, the Linker transmits the query results to the sender of the query.
The whole process is depicted in Figure 3.

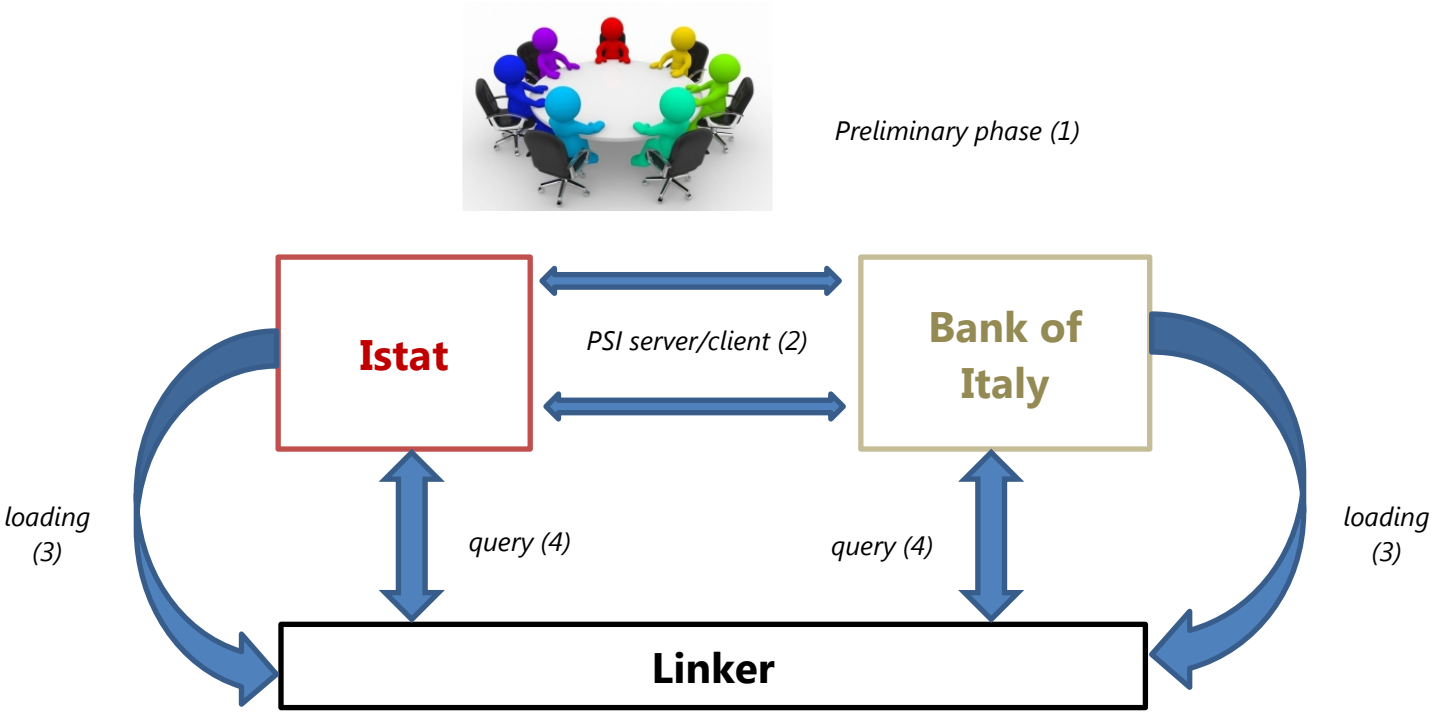

FIGURE 3: EXCHANGE OF INFORMATION BETWEEN THE PARTIES ACCORDING TO THE PROTOCOL

In this paper we have considered some techniques to carry out data linkage among data bases belonging to institutions while preserving a desirable level of anonymity. We have shown some examples of empirical applications with the use of synthetic data.

The main findings of these experiments are the following:

1) Assuming an HbC environment (i.e. a trustful behavior), it is possible to address the data sharing goal between institutions in a *private* framework: each party will know either counts with respect to a given set of grouping variables or the actual values of the attributes of records belonging to the other party, with the privacy constraints enforced on identifier fields.
2) In situations with rarefied distribution of record attributes it could be required the employment of Statistical Disclosure control techniques to assess the risk of reidentification either on the Linker or in the client/server side.

3) The extensions of statistical analyses in the PSI-A framework could be explored with homomorphic cryptography[6] that has recently received a considerable attention in the literature (see e.g. [5]).

On the basis of the empirical applications carried out so far, we think that the field of private information sharing has potential for further investigation and adoption in concrete data sharing scenarios among institutions.
As discussed in the introduction, both technical and organizational actions are suggested to make the result of this work able to be concretely implemented within national and international institutions.

[6] Homomorphic Encryption is a special kind of encryption scheme that allows any third party to operate on the encrypted data without decrypting it in advance.

References

[1] *Rakesh Agrawal, Alexandre V. Evfimievski, Ramakrishnan Srikant: Information Sharing Across Private Databases. SIGMOD Conference 2003: 86-*

[2] Monica Scannapieco, Ilya Figotin, Elisa Bertino, Ahmed K. Elmagarmid: Privacy preserving schema and data matching. SIGMOD Conference 2007: 653-664

[3] Yehuda Lindell , Benny Pinkas, Privacy Preserving Data Mining, Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology, p.36-54, August 20-24, 2000

[4] Emiliano De Cristofaro and Gene Tsudik, Practical Private Set Intersection Protocols with Linear Computational and Bandwidth Complexity, Proceedings of Financial Cryptography and Data Security, 2010

[5] A. ACAR, H. AKSU, and A. S. LUAGAC, M. CONTI A Survey on Homomorphic Encryption Schemes: Theory and Implementation, 2017, https://arxiv.org/abs/1704.03578

[6] R.L Rivest, A. Shamir, L. Adleman. "A method for obtaining a digital signature and public key cryptosystem". Communications ACM, 21(2) 1978, pp. 120-126.

[7] W. Diffie, M.E. Hellman. "New directions in cryptography" IEEE transaction on Information Theory". Vol. IT-22 No 6, 1976, pp. 644-654.

Appendix

The employed cryptographic functions

RSA encryption

The RSA is an algorithm for encryption and digital signature introduced by Rivest, Shamir and Adleman at the MIT in 1977. It provides an asymmetric encryption based on two (public and private) keys distributed between the sender and the receiver.

The RSA algorithm can be split into the following the phases:

1) Key generation;
2) Message encryption;
3) Message decryption.

The key generation consists in:

1) choosing two huge prime numbers (~300 decimal digits) and computing their product $n = p \cdot q$;
2) Compute the value $\lambda(n) = lcm(\lambda(p) \cdot \lambda(q)) = (p-1) \cdot (q-1)$ which is the maximum number of values a coprime with n;
3) Choose an integer e satisfying the conditions $1 < e < \lambda(n)$ and that e and $\lambda(n)$ are coprime;
4) Determine d in such a way that $d \cdot e \equiv 1 \, mod \, \lambda(n)$; i.e., d is the modular multiplicative inverse of e (modulo $\lambda(n)$)

The public key consists of the modulus n and the encryption exponent e; the private key consists of the decryption exponent d and $\lambda(n)$).

At this point the encryption and decryption steps are given by the following modular exponentiation:

encryption: $c \equiv (m)^e \, mod(n)$

decryption: $m \equiv (m^e)^d \equiv m^{e \cdot d} \equiv m^1 \, mod(n)$

Hash functions

Cryptographic hash functions are an essential building block

for privacy and security applications.

Cryptographic hash functions map input strings of arbitrary length to short,

fixed length output strings. They were introduced in cryptology in the 1976 seminal paper of Diffie and Hellman on public-key cryptography [7]. Hash functions can be used in a broad range of applications: to compute a short unique identifier of a string (e.g. for a digital signature), as one-way function to hide a string (e.g. for password protection and for privacy purposes).

In the empirical application here described we have employed the hash functions belonging to the SHA-2 (Secure Hash Algorithm) family. This family of algorithms has been designed by the U.S. National Security Agency (NSA).

A good hash function H should satisfy the following requirements:

1) *One-wayness*: for an arbitrary *n*-bit length string w it is hard to find a value of x such that $H(x) = w$;
2) *Second preimage resistant*: for an arbitrary *n*-bit length string x it is hard to find a value $y \neq x$ so that $H(x) = H(y)$;
3) *Collision resistance*: it is hard to find two values x and y satisfying $y \neq x$ so that $H(x) = H(y)$;

It is straightforward to see that collision resistance implies second-preimage resistance. In practice, collision resistance is the strongest property of all three, hardest to satisfy and easiest to breach, and breaking it is the goal of most attacks on hash functions.

In our implementation we have taken the hash functions present the Python library HASHLIB.