On Accuracy, Privacy and Complexity in the Identification Problem

F. Beekhof, S. Voloshynovskiy, O. Koval and T. Holotyak

1. INTRODUCTION

The identification problem in modern applications such as biometrics and brand protection is concerned with new requirements to privacy and runtime-complexity besides the classical demand for accuracy or the performance in terms of identification error. Hence, previous work is extended by coupling the existing model of identification systems to a model from literature that introduces relevant entities and their interaction; and the introduction of new and more accurate decoders. Besides an evaluation of the accuracy and runtime-complexity of the decoders, the concept of privacy is developed and the privacy protection offered by different types of decoders is then analysed using the interaction model.

Given the attention to privacy, a clear definition is desirable; privacy is the ability of a certain owner of data to reveal parts of that data to others or to keep data secret as desired. Therefore, in our formulation, we will consider the privacy leak as the amount of mutual information between the original data and its representation needed for identification with a specified accuracy.

1.1 Entity Interaction Model

To investigate the privacy issue, we need to define certain roles in a similar spirit as in existing literature.[2] Note that a division in roles is not necessarily identical to a division in people or machines; it is a logical division rather than physical. The different entities and their interactions are depicted in Figure 1.

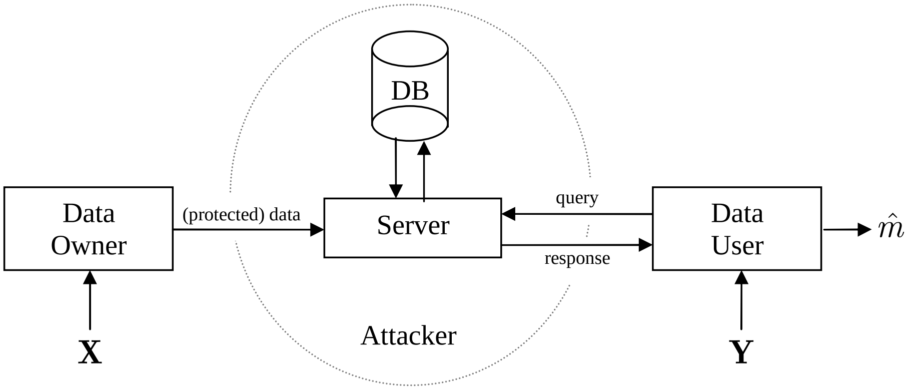

Figure 1. The model of the entities and the interaction.

The *data owner* controls the source of private information $\mathbf{X}$. The main concern relevant to the privacy of the data owner is that the system should not reveal more information that the data owner wishes to disclose.

The data owner passes some protected data to the *server*, which stores it in a database and can perform calculations on the data if desired. The server is assumed to operate properly in the sense that any desired processing will be done correctly, but it is also assumed to be transparant to all.

The *data user* is a trusted entity that is to carry out the identification and is given noisy observations $\mathbf{Y}$ of the original data $\mathbf{X}$ in order to accomplish this task. The output of the identification procedure is the estimated identity $\hat{m}$. The data user can communicate with the server or ask it to perform certain calculations, but can also perform calculations locally. In the latter case, communication with the server in the form of queries and responses are needed to obtain the necessary information.

The *attacker* is a hostile entity whose goal it is to gain knowledge of the private information $\mathbf{X}$ based on everything that the attacker can observe. The attacker can see all information in the database and monitor all calculations in the server as well as all communication between the different entities, i.e., all data disclosed by the data owner as well as the queries and responses that are exchanged between the server and the data user.

1.2 Information Processing Model

In line with previous work,[8] the model of processing information in the system proceeds as shown in Figure 2. The data owner has some private data and needs to process it in order to come to a set of data that will be disclosed. This stage is generally known as the *enrollment*. The source of the original private data is modelled by a continuous memoryless source $\mathbf{X}$ with some distribution $p(\mathbf{x})$. One of the differences with classical communication setups is that the distribution $p(\mathbf{x})$ is given rather than chosen.

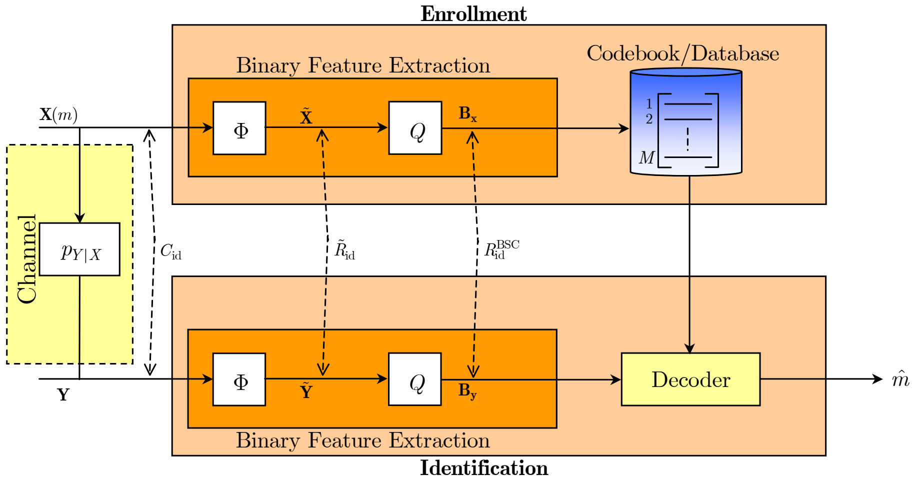

Figure 2. A schematic overview of a framework for privacy-preserving identification.

During the enrollment stage, the dimensionality of the data is reduced from N to $L < N$ to extract a so-called *template*. The reduction is accomplished by applying random projections;[6] we use an approximation of a so-called *orthoprojector* $\mathbf{\Psi}$, where each $\Psi_{i,j} \sim \mathcal{N}(0, \frac{1}{N})$. The dimensionality reduction step transforms $\mathbf{x}$ into $\tilde{\mathbf{x}}$. Then, binary data $\mathbf{b_x}$ is extracted from $\tilde{\mathbf{x}}$ by taking the signs of the projections. That $\mathbf{b_x}$ is disclosed by the data owner by sending it to the server, thus completing the enrollment.

The second stage is the *verification*, and is executed by the data user. The data user can process the received data $\mathbf{y}$ in a similar manner as the data owner to obtain $\tilde{\mathbf{y}}$ and $\mathbf{b_y}$. It is then up to the data user to complete the identification with this information and any communication with the server. The decoder algorithm employed by the data user determines the complexity and privacy leak.

An inherent assumption of the application is the presence of noise in the data $\mathbf{Y}$ that is received by the data user, that should be considered as a degraded version of the original private data $\mathbf{X}$. The probabilistic mismatch between $\mathbf{X}$ and $\mathbf{Y}$ is modelled by the channel $p(\mathbf{y}|\mathbf{x})$. The identification capacity C_{id}[10] is $I(\mathbf{X};\mathbf{Y})$ where $I(.;.)$ is the mutual information. As a consequence of the projections, the achievable rate reduces to $I(\tilde{\mathbf{X}};\tilde{\mathbf{Y}})$.[8,9] The equivalent channel between the binary vectors $\mathbf{B_x}$ and $\mathbf{B_y}$ follows the Binary Symmetric Channel (BSC) model, resulting in an achievable rate of $I(\mathbf{B_x};\mathbf{B_y})$.[3]

1.3 The Identification System

Combining the two presented models, the final system is then characterized by the following configuration: $\mathbf{X}$ is the private data possesed by the data owner, and $\mathbf{B_x}$ is the released information that is communicated to the server and stored there. Let the communication between the data user and the server be denoted as $\mathbf{A}$, then the attacker has access to $\mathbf{B_x}$ and $\mathbf{A}$. The type of the communicated data $\mathbf{A}$ depends on the decoding algorithm.

A more strict definition of privacy can be formulated as the uncertainty that the attacker has about the private information $\mathbf{X}$ given $\mathbf{B_x}$ and $\mathbf{A}$, which can be expressed mathematically as $h(\mathbf{X}|\mathbf{B_x},\mathbf{A})$, where $h(.)$ stands for the differential entropy. This definition is closely linked to the definition of privacy leakage as the mutual information $I(\mathbf{X};\mathbf{B_x},\mathbf{A})$, similar to definitions in literature.[7] The link between the privacy and the privacy leak can be expressed using Shannon's equivocation $I(\mathbf{X};\mathbf{B_x},\mathbf{A}) = h(\mathbf{X}) - h(\mathbf{X}|\mathbf{B_x},\mathbf{A})$.

It can be derived that $h(\mathbf{X}|\mathbf{B_x},\mathbf{A}) \geq h(\mathbf{X}) - h(\mathbf{B_x},\mathbf{A})$, with equality in the case that $h(\mathbf{B_x},\mathbf{A}|\mathbf{X}) = 0$.

2. DECODERS

2.1 Exhaustive Search v.s. Shooting

Exhaustive search decoders simply evaluate each entry in the codebook with respect to the channel output according to any approximation of a maximum-likelihood rule and therefore all have a time-complexity of $O(ML)$. When the data user employs an exhaustive search algorithm, the communicated data $\mathbf{A}$ is simply equal to binary codebook $\mathbf{b_x}$, which is sent from the server to the data user.

An alternative decoder design is based on heuristic-guided backtracking algorithms that try to find the most likely match in the binary codebook by flipping a fraction of the L bits starting with the least reliable ones, inspired by DPLL-solvers for the Satisfiability problem.[4,5] This covers the most likely original codewords and is therefore an approximation of an ML-decoder. The depth of the tree that is expanded during the recursive search is limited to a number $D \leq L$. For this type of decoders, the communicated data $\mathbf{A}$ consists of queries in the form of bitpatterns derived from $\mathbf{b_y}$ and responses indicating whether or not that bitpattern exists in the codebook $\mathbf{b_x}$ stored at the server.

In the final version of this paper, more details of these approaches will be discussed, pointing out the difference in runtime complexity and accuracy.

2.2 Hard v.s. Soft Decoding

The optimal maximum-likelihood decoder for the BSC reduces to a minimum Hamming distance decoder. Let $d_H(.,.)$ denote the Hamming distance between two binary sequences, then the minimum Hamming distance decoder is:

$$\hat{m} = \arg \min_{1 \leq m \leq M} d_H(\mathbf{b_x}(m), \mathbf{b_y}).$$

In the final version of this paper, a different decoding rule will be introduced that is based on $\mathbf{b_x}$ and $\tilde{\mathbf{y}}$, extending previous work,[1] that leads to superior accuracy compared to hard decoders such as the minimum Hamming distance decoder. The soft decoding rules require the same kind of data to be communicated between the data user and the server as long as the soft information remains on the side of the data user; soft decoders also have the same complexity as hard decoding. Using this result, we will show in the Section 3 that this improved accuracy does not lead to loss in privacy.

3. PRIVACY ANALYSIS

We will show that for the chosen decoders the privacy leak is due only to the data revealed during enrollment. Under the chosen assumptions that the attacker has access to $\mathbf{B_x}$ and $\mathbf{A}$, we will show that the privacy is $h(\mathbf{X}|\mathbf{B_x},\mathbf{A}) \geq h(\mathbf{X}) - h(\mathbf{B_x},\mathbf{A}) = h(\mathbf{X}) - h(\mathbf{B_x})$ for both the exhaustive and shooting type of decoders. In other words, in both cases there is no privacy leak other than the one chosen by the data owner.

Other than the disclosed data $\mathbf{B_x}$, a second source of information for the attacker can be the communication between the server and the data user $\mathbf{A}$. Under the formulation of the problem as presented in Section 1.1 stating that the attacker has no access to any information available to the data user other than the communication with the server, and with the decoders described in Section 2 running on the side of the data user, the attacker cannot gain anything beyond what is already available in the form of $\mathbf{B_x}$.

The only data communicated between the server and the data user for exhaustive search decoders is the codebook $\mathbf{B_x}$ which is already known to the attacker. Thus, for exhaustive search decoders $\mathbf{A} = \mathbf{B_x}$, from which it follows that $h(\mathbf{X}|\mathbf{B_x},\mathbf{A}) = h(\mathbf{X}|\mathbf{B_x},\mathbf{B_x}) = h(\mathbf{X}|\mathbf{B_x})$.

For the shooting type of decoders, the communication $\mathbf{A}$ consists of $\mathbf{B_y}$ and modified version thereof, some of which may be valid entries in the codebook; the responses are in this case indicators showing if such an entry is in the codebook or not. Given that $\mathbf{B_y}$ is a degraded version of $\mathbf{B_x}$, it can be expressed as a function of $\mathbf{B_x}$, i.e. $\mathbf{B_y} = g(\mathbf{B_x})$, so that $h(\mathbf{X}|\mathbf{B_x}, \mathbf{A}) = h(\mathbf{X}|\mathbf{B_x}, \mathbf{B_y}) = h(\mathbf{X}|\mathbf{B_x}, \mathbf{g}(\mathbf{B_x})) = h(\mathbf{X}|\mathbf{B_x})$. Hence, the attacker cannot gain any more accurate information about $\mathbf{X}$ from $\mathbf{B_y}$ than from $\mathbf{B_x}$.

In conclusion, the attacker cannot learn anything about the private information $\mathbf{X}$ other than that which has been disclosed by the data owner. What is more important however, is the fact that the information disclosed to the attacker is the same for hard and soft decoding strategies if the calculation is executed on the side of the data user. Therefore, the superior accuracy of soft decoders can safely be exploited without sacrifice to either privacy or complexity.

4. PERFORMANCE ANALYSIS

In the final version of this article, we will evaluate the performance and complexity for varying SNRs of different decoders for which we have considered the capabilities of an attacker as outlined in Section 3.

Acknowledgments

This work is supported by SNF projects 111643 and 1119770.

REFERENCES

1. F. Beekhof, S. Voloshynovskiy, O. Koval, and T. Holotyak. Fast identification algorithms for forensic applications. In *Proceedings of First WIFS*, 2010.
2. R. Brinkman. *Searching in Encrypted Data.* PhD thesis, University of Twente, 2007.
3. T. Cover and J. Thomas. *Elements of Information Theory.* Wiley and Sons, New York, 1991.
4. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962.
5. M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.
6. J. Fridrich. Robust bit extraction from images. In *Proceedings ICMCS'99*, volume 2, pages 536–540, Florence, Italy, June 1999.
7. T. Ignatenko. *Secret-Key Rates and Privacy Leakage in Biometric Systems.* PhD thesis, Technical University of Eindhoven, 2009.
8. S. Voloshynovskiy, O. Koval, F. Beekhof, and T. Pun. Conception and limits of robust perceptual hashing: toward side information assisted hash functions. In *Proceedings of SPIE Photonics West, Electronic Imaging / Media Forensics and Security XI*, San Jose, USA, 2009.
9. S. Voloshynovskiy, O. Koval, and T. Pun. Multimodal authentication based on random projections and distributed coding. In *Proceedings of the 10th ACM Workshop on Multimedia & Security*, Oxford, UK, September 22–23 2008.
10. F. Willems, T. Kalker, J. Goseling, and J-P. Linnartz. On the capacity of a biometrical identification system. In *In: Proc. of the 2003 IEEE Int. Symp. on Inf. Theory*, pages 8–2, 2003.