Privacy Preserving ECG Quality Evaluation

Riccardo Lazzeretti
Department of Information Engineering
University of Siena
Siena, Italy
riccardo.lazzeretti@gmail.com

Jorge Guajardo*
Philips Research Europe
Eindhoven
The Netherlands

Mauro Barni
Department of Information Engineering
University of Siena
Siena, Italy
barni@dii.unisi.it

ABSTRACT

Remote health-care applications are gaining popularity as an alternative for patients who do not require hospitalization. In this setting, privacy preserving protocols are useful to enable the offering of personalized online services, thus preventing the unnecessary disclosure of personal data. A problem often neglected in privacy-preserving protocols is the need to ensure that processed signals, which are often recorded by non-expert consumers, are of sufficient quality, hence raising the need for solutions that assess the quality of the recorded signals to guarantee correct (medical) decisions. In this paper, we propose a privacy preserving protocol that assesses signal quality and combines this with a linear classifier used to decide whether the measured signal is of high enough quality or not. In particular, the protocol computes a frame based Signal-To-Noise Ratio (SNR) from the original signal and a filtered version of the signal itself; evaluates the mean and the variance of the SNRs obtained and computes the overall signal SNR. Finally these measures are combined with a linear classifier used to assess the quality of the signal. The proposed scheme relies on a hybrid multi-party computation protocol based on Homomorphic Encryption and Yao's Garbled Circuits. The analysis of the protocol indicates that it needs the transmission of less than 4 MBytes of data to analyze 30 seconds of ECG signals providing a classification accuracy close to 85%.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems; G.1.6 [**Numerical Analysis**]: Optimization—*Integer programming*; J.3 [**Life and Medical Sciences**]: Health; k.4.1 [**Computers and Society**]: Public Policy Issues—*Computer-related health issues,Privacy*

General Terms

Algorithms, Security

Keywords

ECG Quality Evaluation, Signal Processing in the Encrypted Domain, Privacy Preserving Solutions, Garbled Circuits, Homomorphic Encryption, SNR, Classification

*The author is currently with Robert Bosch LLC, Pittsburgh, U.S.A.

1. INTRODUCTION

The health-care industry is moving faster than ever towards technologies offering personalized online self-services, medical error reduction, customer data collection and more. Such technologies have the potential to revolutionize the way medical data is managed, stored, processed, delivered and ubiquitously made available to millions of users throughout the world. Such service offerings raise security and privacy issues. On the one hand, the service provider (a company or an hospital) may not be willing to provide the end user directly with its proprietary protocols because of fear of disclosing valuable intellectual property to third parties or compromising the basis for its service. On the other hand, patients or users of the service might not be willing to disclose their personal signal information to a third party (not necessarily a hospital or doctor) for fears of losing control of their data (e.g., a user might be concerned that due to certain medical condition his insurance premium will increase or that he will be discriminated during a job application).

This paper addresses one of the basic safety problems in the domain of tele-health, which is the problem of guaranteeing the recorded signals comply with certain quality measures. Namely, when data remotely measured by patients is used by tele-health services or in the medical professional world, healthcare providers need to place greater trust in the information that patients report. In particular, in order to make sound medical decisions based on this remotely measured biomedical signals, they have to be ensured that a measurement contains as little noise as possible, the amount of noise being an indication of the quality of the signal. This is very important because if this is not guaranteed there can be critical health care decisions made based on wrong (or poor quality) data, which in turn, can lead to wrong decisions or treatments. As a particular relevant application, we consider privacy-preserving quality evaluation of electrocardiograms (ECGs).

1.1 Electrocardiograms

In this section, we provide an introductions to the electrocardiogram and its properties. The interested reader is referred to [1] for a more detailed treatment of the ECG.

Heart contractions are due to an elaborate electrical conduction system that controls the precise timing for depolarizing the substantial mass of the electrically excitable myocardium. The graphical recording of electrical heart signals, named Electrocardiogram (ECG), is obtained by using surface electrodes attached to the skin in such a way that different sections of the heart are crossed. The placement of the electrodes determines the directional viewpoint of the heart. A single-lead ECG recorder would typically have three electrodes: the positive electrode, the negative electrode and an indifferent electrode (ground or right-leg drive electrode). An ECG signal usually assumes values between -5mV and 5mV and the number of bits used for its representation depends on the accuracy of the electrocardiograph.

A typical ECG tracing of the heartbeat consists of a P wave (corresponding to the atrial depolarization), a QRS complex (reflecting the rapid depolarization of the right and left ventricles), a T wave (showing the ventricles repolarization) and some other smaller waves. Observing the slope of these waves and the segments connecting them, a cardiologist can derive a lot of information (for example the heart rate) or identify diseases, such as Arrhythmias, ischemia, injury and infarction. Many other diseases can be identified by observing or analyzing the ECG as described in [1]. Moreover many protocols have been proposed to classify ECG between Normal Sinus Rhythm and some arrhythmias. One of them has been also implemented in the encrypted domain [10].

In hospitals, electrocardiograms are normally recorded in the presence of an expert (nurse, doctor or technician), but in a remote monitoring scenario, the patient is responsible for performing such measurements without the help of a specialized care provider. The recorded ECG can be later analyzed by a care provider or used as input to a (remote) application, which performs some analysis of the data and outputs a result or advice [10]. Advantages of such a service include allowing the user to check regularly its health state without going to an analysis center or hospital. This translates in cost reductions for the service provider and convenience for the patient, since monitoring is performed in the comfort of his home. However, one may ask what happens if the signal sent to the service provider for analysis is affected by noise. In this case, the service provider might reach the wrong conclusion or provide the user with the wrong advice leading to unnecessary worries and costs.

ECG signals can be contaminated by noise due to different factors and with different power. The main sources of noise are:
Power Line Interference, which consists of 50/60 Hz pickup and harmonics, depending on the country power line.
Electrode contact noise, which is a transient interference caused by loss of contact between the electrode and the skin and can be permanent or intermittent. The switching action can result in large artifacts (i.e., changes in the ECG due to external noise) since the ECG signal is usually capacitively coupled to the system.
Motion artifacts, which are transient base line changes in the electrode skin impedance produced by electrode motion.
Muscle contraction, which causes generation of artifactual millivolt level potentials.
Baseline wander, which in ECG signals causes problems in the detection of peaks.

Power line interference and baseline wander can be easily removed with the filter proposed in [21]. Motion artifacts and muscle contraction depend on the patient and in general can not be avoided. On the other hand, electrode contact noise is the only type of noise that can be prevented with correct placement of electrodes. Thus, in remote monitoring scenarios in which electrodes are placed by non-expert people, control of the recorded signal quality becomes very important.

1.2 Our Contributions

In this paper, we describe a methodology which allows the service provider to obtain a quality measure so as to guarantee that the input to later computations and the corresponding output (the actual analysis performed on the data) comply with certain confidence criteria. In particular, we propose a protocol that allows evaluating the quality of ECG signals and communicates to the user whether the input signal is good enough to continue with the measurement and analysis or not. In the negative case, a new measurement is requested and subsequent re-sending of the signal is required. The protocol can be easily implemented in a privacy preserving manner that permits the client $\mathcal{C}$ not to reveal his data to the server $\mathcal{S}$, yet the server is able to perform the analysis of the ECG quality without revealing the private parameters (e.g., the filter) of its algorithm. Both the client and the server can be interested to obtain as much information as possible, but they are not interested to deviate from the protocol, being the patient's health involved, hence the semi-honest model is considered.

Our solution is novel especially from a signal processing and privacy preserving techniques points of view. In particular, we propose a new technique to analyze the amount of noise in a biomedical signal based on analysis of the Signal-to-Noise ratio in small windows of the encrypted signal rather than the whole measurement [5]. The analysis is based on the statistics of the raw signal window and a corresponding filtered signal. To our knowledge no alternative solutions have been proposed in the literature. In addition, we report on the performance of a simple linear classifier on the quality measure, which achieves close to 85% accuracy.

The remainder of this paper is organized as follows. Section 2 introduces the signal processing and cryptographic tools we will use in the following sections to develop our solutions. The ECG quality estimation algorithm is presented in Section 3, together with its privacy preserving version. In Section 4, we analyze the dimension of the data involved in the computation and the communication complexity of the protocol, moreover the results of experiments conducted on ECG signals from the MIT-Arrhythmia database in terms of classification accuracy are presented. We end with some conclusions in Section 5.

2. TOOLS

In this section, an overview of the cryptographic primitives and signal processing tools used in the paper is provided.

2.1 Cryptographic Tools

Homomorphic Encryption (HE). Out protocol is based on a semantically secure additively homomorphic public-key encryption scheme. In an additively homomorphic cryptosystem, given encryptions $[\![a]\!]$ and $[\![b]\!]$ (where $[\![\cdot]\!]$ indicates the encrypted version of a and b, respectively), an encryption $[\![a+b]\!]$ can be computed as $[\![a+b]\!] = [\![a]\!][\![b]\!]$, where all operations are performed in the corresponding plaintext or ciphertext domain. From this property, it follows that multiplication of an encryption $[\![a]\!]$ with a constant c can be computed efficiently as $[\![c \cdot a]\!] = [\![a]\!]^c$. As instantiation of an additively homomorphic encryption scheme, we use the Paillier cryptosystem [20, 9], which has plaintext space $\mathbb{Z}_N$ and ciphertext space $\mathbb{Z}^*_{N^2}$, where N is a T-bit RSA modulus. For details on the Paillier encryption and decryption functions we refer to [20, 9].

Parallel Oblivious Transfer (OT). A parallel 1-out-2 Oblivious Transfer for ℓ strings of bitlength t, denoted as OT^ℓ_t, is a two-party protocol where $\mathcal{S}$ inputs ℓ pairs of t-bit strings $S_i = \langle s_i^0, s_i^1 \rangle$ for $i = 1, \ldots, \ell$ with $s_i^0, s_i^1 \in \{0,1\}^\ell$ and $\mathcal{C}$ inputs ℓ choice bits $b_i \in \{0,1\}$. At the end of the protocol, $\mathcal{C}$ learns $s_i^{b_i}$, but nothing about $s_i^{1-b_i}$ whereas $\mathcal{S}$ learns nothing about b_i. We use OT^ℓ_t as a black-box primitive in our constructions. It can be instantiated efficiently with different protocols [19, 2, 17, 12]. In this paper we consider the protocol described in [19], which - when implemented over a suitably chosen elliptic curve - has asymptotic communication complexity $6\ell t$ and is secure against malicious $\mathcal{C}$ and semi-honest $\mathcal{S}$ in the random oracle model. Extensions of [12] can be used to reduce the number of computationally expensive public-key operations to $\approx 6t^2 + 4\ell t$ and is used when $\ell > 3t$. Moreover OT can be precomputed [6], performing an offline OT on random values that is later used in the online OT phase to obtain the correct result from the actual input values with asymptotic complexity $2\ell t$ bits.

Garbled Circuit (GC). Yao's Garbled Circuit approach [23], excellently presented in [16], is the most efficient method for secure evaluation of a boolean circuit C in the two party setting. We summarize the main ideas in the following. First, the circuit **constructor** (server $\mathcal{S}$), creates a *garbled circuit* $\widetilde{C}$: for each wire W_i of the circuit, he randomly chooses a *complementary garbled value* $\widetilde{W}_i = \langle \widetilde{w}_i^0, \widetilde{w}_i^1 \rangle$ consisting of two secrets, $\widetilde{w}_i^0$ and $\widetilde{w}_i^1$, where $\widetilde{w}_i^j$ is the *garbled value* of W_i's value j. (Note: $\widetilde{w}_i^j$ does not reveal j.) Further, for each gate G_i, $\mathcal{S}$ creates and sends to the **evaluator** (client $\mathcal{C}$) a *garbled table* $\widetilde{T}_i$ with the following property: given a set of garbled values of G_i's inputs, $\widetilde{T}_i$ allows to recover the garbled value of the corresponding G_i's output, and nothing else. Then garbled values corresponding to $\mathcal{C}$'s inputs x_j are (obliviously) transferred to $\mathcal{C}$ with a parallel oblivious transfer protocol OT: $\mathcal{S}$ inputs complementary garbled values $\widetilde{W}_j$ into the protocol; $\mathcal{C}$ inputs x_j and obtains $\widetilde{w}_j^{x_j}$ as outputs. Now, $\mathcal{C}$ can evaluate the garbled circuit $\widetilde{C}$ to obtain the garbled output simply by evaluating the garbled circuit gate by gate, using the garbled tables $\widetilde{T}_i$. Correctness of GC follows from the method of construction of the garbled tables $\widetilde{T}_i$. As in [8], we use the GC protocol as a conditional oblivious transfer protocol where we do not provide a translation from the garbled output values to their plain values to $\mathcal{C}$, i.e., $\mathcal{C}$ obtains one of two garbled values which can be used as key in subsequent protocols but does not know the value key corresponds to.

High-Speed evaluation of GC [18] is feasible by using a cryptographic hash function $H(\cdot)$ (chosen from the SHA-2 family). The creation of the garbled table associated to a d-input gate requires 2^d invocations of $H(\cdot)$. A *point-and-permute technique* can be used to speed up the implementation of the GC protocol [18]: the garbled values $\widetilde{w}_i = \langle k_i, \pi_i \rangle$ consist of a symmetric key $k_i \in \{0,1\}^t$ and $\pi_i \in \{0,1\}$ is a random permutation bit. The permutation bit π_i is used to select the right table entry for decryption with the key k_i, hence only one invocation of $H()$ for each table is needed during evaluation. The *free-XOR* gates technique introduced in [13], can be used to further improve the performance of the GC technique, so that XOR gates need not be created nor their corresponding garbled tables transmitted and evaluation is performed by a simple XOR operation. The output of the GC is converted to plain values by using a two rows conversion table for each output bit.

2.2 Signal Processing Tools

Filters. A filter is represented in the spectral domain by the function

$$H(z) = \frac{num(z^{-1})}{den(z^{-1})},$$

where $num()$ and $den()$ are polynomials, and can be implemented in the time domain by a difference equation. Filters can be subdivided into two categories: Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters.

An FIR filter is characterized only by a numerator $num(z^{-1}) = c_0 + c_1 z^{-1} + \ldots + c_k z^{-k}$ of order k. Each filtered sample y_i is computed by a scalar product between an array composed by the current and the previous k samples $\{x_i, \ldots x_{i-k}\}$ and the coefficients array $\{c_0, \ldots c_k\}$ of the numerator ($y_i = c_0 x_i + c_1 x_{i-1} + \ldots + c_k x_{i-k}$). The filter introduces a delay of $k/2$ samples that can be avoided in non-real time applications by computing $y_i = c_0 x_{i+k/2} + \ldots + c_{k/2-1} x_{i+1} + c_{k/2} x_i + c_{k/2+1} x_{i-1} + \ldots + c_k x_{i-k/2}$. Moreover if the filter is real (has no imaginary components), its spectrum is symmetric with respect to the frequency $f = 0$ Hz and the coefficients are symmetric with respect to the central coefficient c_0 and as a result $c_{+j} = c_{-j}\ \forall j = 1..k/2$. We can easily describe a real filter with the difference equation $y_i = c_0 x_i + \sum_{j=1}^{k/2} c_j (x_{i+j} + x_{i-j})$, where $c_j = n_{k/2+j} = n_{k/2-j}$. A filter having floating point coefficients assures greater accuracy than a filter with fixed-point (integer) coefficients. However filter having fixed point coefficients can be more easily implemented in the encrypted domain.

An IIR filter has a numerator of k_n order and a denominator of k_d order and the i-th filtered sample is computed as $y_i = -d_1 y_{i-1} - \ldots - d_{k_d} y_{i-k_d} + c_0 x_i + c_1 x_{i-1} + \ldots + c_{k_n} x_{i-k_n}$

by using the last filtered samples previously obtained, together with the samples of the original signal. Because the previous filtered samples use other samples, in a recursive procedure, the current filtered sample is computed as a function of all the previous samples. An IIR filter normally requires less coefficients than an FIR filter, but an integer implementation of the IIR filter usually amplifies the filtered signal. Thus, we avoid divisions, and the IIR filter amplifies the filtered signal exponentially. On the other hand, known protocols for division in the encrypted domain are expensive [22, 14] and the lower complexity obtained by decreasing the number of coefficients is more than compensated by the overhead introduced to perform a division immediately after the computation of every filtered sample. Thus, in general an FIR filter is more efficient than an IIR filter in the encrypted domain.

3. PROTOCOL

Different parts of the ECG carry different types of information. For example, to evaluate the heart rate or some arrhythmias, it is sufficient to be able to identify the R peaks (the high peaks of the QRS complexes), for the atrial flutter we need to observe if there is the "saw-tooth" effect and to identify an ischemia it is necessary to evaluate the slope of the ST segment and of the T wave.

An ECG signal can have enough quality to allow the evaluation of a particular function or the extraction of particular information, it can have bad quality not allowing information extraction, or sufficient quality for certain analysis/applications but not sufficient for others. Thus, the quality of an ECG signal is application-dependent.

It is important to stress that quality evaluation techniques can be split into two main categories: full reference or no-reference techniques. The former class refers to a situation in which the quality of a signal has to be judged by referring to an ideal signal. This is the case in lossy compression applications wherein the quality of the compressed signal has to be judged by considering its perceptual distance from the original signal. In contrast, no-reference techniques have to measure the quality of a signal without making any reference to an ideal signal that is supposed to represent the maximum possible quality. The medical scenarios addressed in this paper require that no-reference quality measures are adopted.

In this paper, we propose a system that evaluates ECG signal quality by identifying the presence of electrode contact noise. Our purpose is to provide a solution allowing a simple implementation in the encrypted domain, even if not optimal from the accuracy point of view. Evaluating the quality of a non-referenced signal is a difficult problem for which only a few solutions are known even in the plain domain (see [15, 3] for example). Existing solutions, however, are generally too complex to allow an efficient implementation in the encrypted domain.

In [5] a garbled circuit is proposed to evaluate the SNR of a signal. Clearly, given a clean and a noisy version of a signal, the signal SNR can be computed. However, a person interested in evaluating the SNR of an ECG signal faces several challenges. The first one is that in traditional (tele-monitoring) applications the real *clean* ECG signal is not available. Rather, only a noisy version is available. The signal can be filtered by using a filter that removes all the frequencies outside the interval where the signal power is expected to be. The filtered signal is considered an estimation of the clean signal, while the recorded signal is the noisy signal. In [5], the authors notice that the larger the estimated noise in the recorded signal, the larger amount of noise the filter will remove, even if the filtered signal will still be affected by noise. Hence the SNR between the filtered signal and the noise signal can be used to estimate if noise is also affecting the filtered signal.

Unfortunately problem is that generally the correlation of SNR alone with subjective quality is very poor and it is of little interest as a general objective measure of signal quality. The main idea behind the present work is to extend the method proposed in [5] by using a segmented SNR instead of the overall signal SNR. In doing so, we follow the approach proposed in [11], where a frame based segmented SNR is used to assess *speech* signal quality. In particular, the same idea can be applied to many other types of signals having a *fixed range of frequencies* that can be affected by burst of noise such as, videos, ECGs, etc. The method in [11] subdivides the signal into small frames and for each of them the SNR is computed. Finally, the mean and variance of the segmented SNR is computed. The use of the variance is justified by the observation that while the electrode contact noise has a minor impact on the mean SNR, an occasional burst of noise having small time length can be better detected by the SNR variance. Hence both the mean and the variance of the segmented SNR can be used to evaluate the quality of a signal, together with other features (here we consider the SNR of the whole signal) to obtain a more accurate analysis. Notice that in [11], the authors propose additional "plain" techniques to assess the quality of a speech signal, but these would be too computationally intensive to be efficiently implemented in the encrypted domain.

In the following, we describe the steps necessary to compute the SNR in the encrypted domain and reach a decision based on features (e.g. SNR, mean SNR, SNR variance, etc.) extracted during the processing of the signal. The steps are summarized in Figure 1. In the remainder of this paper, assume that we evaluate τ seconds of an ECG signal $\mathbf{x} = \{x_1, \ldots, x_{\tau * f_s}\}$, where f_s is the sampling frequency and each sample is represented with $\ell_x + 1$ bits (ℓ_x bits for the magnitude and 1 for the sign).

Signal Quantization and Representation. To start with we observe that signal processing algorithms usually work on real (floating point) numbers. On the contrary, when working with encrypted signals, it is necessary to represent each value by an integer number, thus, quantizing it. This is equivalent to multiplying the number by a constant and then round it to the closest integer value.

Baseline Wander and Power Line Interference Removal. Quality evaluation starts by removing the base-line wander and the power-line interference by using the filter proposed in [21] producing a filtered signal $\mathbf{y} = \{y_1, \ldots, y_{\tau f_s}\}$. This filtering operation removes noise, which otherwise could be

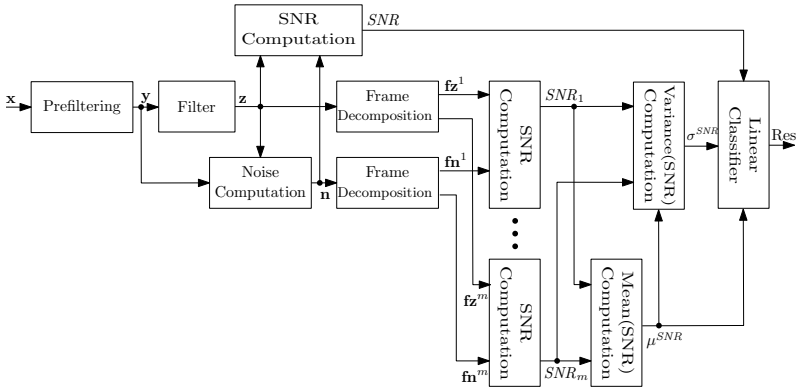


Figure 1: Sequence of steps performed to evaluate the quality of an ECG signal.

incorrectly confused with the electrode contact noise. The filter is developed in the Fourier domain and it has a periodic spectrum with small stop-band notches at 0 Hz as well as at f_p Hz and at its higher harmonics, where f_p is the frequency of the power line. Since the filter is well-known in literature, we assume that it is applied in the plain domain. The client can apply the filter with floating point coefficients on the quantized integer $\mathbf{x}$ signal and then round the value to the closest integer. Filter operations are designed so to not amplify the signal, even if sometime some output values could slightly exceed the maximum value. In particular, output vector components y_i that exceed the maximum representable value are clipped to the maximum value that the electrocardiograph can record. Hence the samples of $\mathbf{y}$ can be represented with the same number of bits as $\mathbf{x}$.

Filtering. According to the American Heart Association recommendations for ECG recordings, in an ECG the typical high frequency component is the QRS complex which is about 20 Hz. To remove high frequency noise, we apply an integer low-pass filter with cut off frequency $f_c = 20$ Hz to the signal $\mathbf{y}$, obtaining the filtered signal $\mathbf{z} = \{z_1, \ldots, z_{\tau f_s}\}$. Assuming that the filter is developed by $\mathcal{S}$ who is interested in protecting his intellectual property, $\mathcal{S}$ applies the filter on the HE encrypted samples transmitted by the client. The reason why HE is preferable to GC for this operation is threefold. First $\mathcal{S}$ can perform all the computation without the need to interact with the client. In contrast using GC requires many product circuits to be transmitted between the client and the server. Thus, the lack of interaction in the evaluation of the FIR filter allows $\mathcal{S}$ to protect the filter order as well as the coefficients. In addition, $\mathcal{S}$ knows the values of the filter coefficients, hence it can optimize the computation avoiding the computation of any products with zero or one coefficients. Moreover, $\mathcal{S}$ can further optimize the filter evaluation by computing once each product between a sample and all filter coefficients having the same value[1].

More formally, $\mathcal{C}$ encrypts the τf_s samples and transmits the encrypted signal $[\![\mathbf{y}]\!] = \{[\![y_0]\!], \ldots, [\![y_{\tau f_s - 1}]\!]\}$ to $\mathcal{S}$. The signal $[\![\mathbf{y}]\!]$ is processed by a filter of order $2k$, having $k+1$ coefficients $c_0 \ldots c_k$, each coefficient is represented with ℓ_c bits. $\mathcal{S}$ computes all the products

$$[\![y_i c_j]\!] = \begin{cases} 1 & \text{if } c_j = 0, \\ [\![y_i]\!] & \text{if } c_j = 1, \\ [\![y_i]\!]^{c_j} & \text{else.} \end{cases}$$

Assuming that the filter has α non-null and non-unitary coefficients, only $\alpha \tau f_s$ products are evaluated. Then, $\mathcal{S}$ obtains each filtered sample by computing $[\![z_i]\!] = [\![c_0 * y_i + \sum_{j=1}^{k}(c_j y_{i+j} + c_j y_{i-j})]\!] = [\![c_0 y_i]\!] \prod_{j=1}^{k}([\![c_j y_{i+j}]\!][\![c_j y_{i-j}]\!])$, where $[\![y_{i \pm j}]\!] = [\![0]\!]\ \forall i + j > \tau f_s$ or $i - j \leq 0$. A final optimization consists in using the packed signal representation proposed in [7]. This technique can be used to pack more samples in a cyphertext, so that communication complexity is reduced.

Energy Evaluation. Let the *noise signal* $\mathbf{n} = \{n_1 \ldots n_{\tau f_s}\}$ be defined as the difference between the signals $\mathbf{z}$ and $\mathbf{y}$. Before going on, we emphasize that filtering a signal in the encrypted domain produces a $\mathbf{z}$ signal amplified by a factor *amp*. Hence, we have to multiply the y samples by the same factor before the subtraction. Each noise sample is hence computed as $[\![n_i]\!] = [\![z_i - amp * y_i]\!] = [\![z_i]\!] * [\![y_i]\!]^{-amp}$.

Then, we subdivide the signals $\mathbf{z}$ and $\mathbf{n}$ into frames of w samples, obtaining $m = \lfloor \tau f_s / w \rfloor$ frames $\mathbf{fz} = \{\mathbf{fz}^1 \ldots \mathbf{fz}^m\}$ and $\mathbf{fn} = \{\mathbf{fn}^1 \ldots \mathbf{fn}^m\}$, where it is assumed that $\tau f_s = mw$. Otherwise, if $\tau f_s \neq mw$, the last frame, having less than w samples, is discarded. For each pair of signal and noise frames $(\mathbf{fz}^i, \mathbf{fn}^i)$ the SNR is evaluated as the power ratio between the signal and the noise in the logarithmic decibel scale:

$$SNR_i^f = 10 \log_{10} \frac{\frac{(\sum_{j=1}^{w}(fz_j^i)^2)}{w}}{\frac{(\sum_{j=1}^{w}(fn_j^i)^2)}{w}} = \frac{10}{\log_2 10} \log_2 \frac{\sum_{j=1}^{w}(fz_j^i)^2}{\sum_{j=1}^{w}(fn_j^i)^2}.$$

Notice that $10/\log_2 10$ only amplifies the result and therefore it can be neglected. Hence the function we need to

[1]We ignore the possible information leakage due to variable computation timings among filters with different number of coefficients.

compute is

$$SNR_i^f = \log_2 \frac{\sum_{j=1}^{w}(fz_j^i)^2}{\sum_{j=1}^{w}(fn_j^i)^2} = \log_2 \frac{E_{fz^i}}{E_{fn^i}}$$
$$= \log_2 E_{fz^i} - \log_2 E_{fn^i},$$

where E indicates the signal energy.

To carry out the previous computation in the encrypted domain, we compute the frames' energy using HE and then obtain the SNR by using GC. Notice that the energy computation protocol has to be applied to each signal frame $\mathbf{fz}^1, \ldots, \mathbf{fz}^m$ and each noise frame $\mathbf{fn}^1, \ldots, \mathbf{fn}^m$. We recall that each frame consists of w samples and the samples are available to $\mathcal{S}$ in their encrypted form.

Then, $\mathcal{S}$ additively blinds each sample s_i by adding a value r_i, i.e. $[\![s_i + r_i]\!] = [\![s_i]\!][\![r_i]\!]$. The values r_i are randomly chosen in $\mathbb{Z}_{\ell+\kappa}$, where ℓ is the bitlength of the sample (in our case ℓ_z or ℓ_n) and κ is a security parameter (usually $k = 80$). At this point $\mathcal{S}$ transmits the ciphertexts containing the obfuscated samples to $\mathcal{C}$, who decrypts them and computes the energy of the obfuscated signal $E_{s,o} = \sum_{i=1}^{w}(s_i + r_i)^2$. To reduce the number of ciphertexts transmitted, $\mathcal{S}$ can pack $\lfloor T/(\ell + \kappa) \rfloor$ obfuscated samples in a single ciphertext.

In a protocol whose goal is computing the energy, $\mathcal{C}$ usually encrypts the values obtained and transmits them to $\mathcal{S}$, who removes the total obfuscation by using the homomorphic property. Considering that the energy values are needed again on $\mathcal{C}$'s side as input to the GC computing the SNR, the protocol can be changed to reduce the number of rounds and the number of transmitted ciphertexts. In particular, $\mathcal{S}$ computes $[\![r]\!] = [\![r' - \sum_i 2r_i * s_i]\!] = [\![r']\!] * \prod_i [\![s_i]\!]^{-2r_i}$ and transmits these values together with the encryption of the obfuscated samples, where r' is a random value chosen in $\mathbb{Z}_{\ell_e+\kappa}$ (ℓ_e is the bitlength of the energy) introduced to avoid leakage of information to the client due to the disclosure of $\sum_i 2s_i r_i$. $\mathcal{C}$, after decryption, adds the value to the obfuscated energy, obtaining $E_{s,o'} = E_s + \sum_i r_i^2 + r'$. In this way the energy is still obfuscated and $\mathcal{C}$ is not able to know its exact value, while $\mathcal{S}$ knows the exact obfuscation value introduced in the energy but not the actual energy value.

The previously described protocol is applied to the $2m$ frames $\mathbf{fz}^1, \ldots, \mathbf{fz}^m$, $\mathbf{fn}^1, \ldots, \mathbf{fn}^m$ (using different random values for each frame). At the end, $\mathcal{C}$ obtains the obfuscated energy for each frame, i.e. $E^1_{fz,o'}, \ldots, E^m_{fz,o'}$, $E^1_{fn,o'}, \ldots, E^m_{fn,o'}$, while $\mathcal{S}$ knows the total obfuscation introduced in the energy of each frame.

SNR Evaluation. The SNR of each frame is computed by using a garbled circuit, where $\mathcal{C}$'s inputs are the obfuscated energies and $\mathcal{S}$'s inputs the obfuscation values.

The boolean circuit first removes the obfuscation affecting the energies. Considering that a subtraction circuit starts the computation from the least significant bit, it is sufficient that $\mathcal{S}$ and $\mathcal{C}$ submit only the ℓ_e least significant bits of each input (ℓ_E bits for the energies relative to the whole signal and their obfuscation values). The garbled circuit implementing the SNR computation is the one described in [5]. Once the obfuscation is removed and the energy of a generic frame i is obtained, the GC computes the SNR of the frame by evaluating the logarithm of the energies E_{fz^i} and E_{fn^i}, which is equivalent to detecting the minimum number of bits necessary to represent the numbers.

Let E be a generic signal energy represented with ℓ_e bits, whose binary representation is denoted by $E_{\ell_e} \ldots E_1$. Notice that the most significant bits of E can be zero. Let a be the position of the most significant non-zero bit of E. From E, one can build a bitstring E' having all the bits from the ℓ_e-th to the $(a+1)-th$ position set to 0 and all the remaining bits set to 1. E' is obtained by setting the most significant bit as E_{ℓ_e} and then each bit of the result E' is computed as $E'_i = E'_{i+1} \vee E_i$, hence $\ell_e - 1$ OR ($\vee$) gates are needed. Instead of computing $\lfloor \log_2 E \rfloor$, the circuit outputs the secrets corresponding to $\lfloor \log_2 E \rfloor + 1$ if E is positive and to 0 if it is null. Counting the bits of E' equal to 1 and subtracting 1 the logarithm result can be obtained.

Given E_{fz^i}, E_{fn^i} one can compute E'_{fz^i} and E'_{fn^i}, which in turn allows one to compute $E'_{fz^i} \oplus E'_{fn^i}$, which contains the number of ones equal to the difference between E'_{fz^i} and E'_{fn^i} or in other words the magnitude of the SNR. Finally, the SNR_i^f is obtained by counting the non-zero bits of $E'_{fz^i} \oplus E'_{fn^i}$. The counter circuit proposed in [5] requires $\approx \ell_e - \log_2(\ell_e + 1)$ non-XOR gates. In this way the magnitude of the SNR is obtained, but not the sign, for which it is sufficient to evaluate $E_{fz^i} < E_{fn^i}$.

Notice that the SNR values are not the final result, they have to be used in later computations and hence kept secret. To avoid their disclosure to $\mathcal{C}$, the GC finally blinds each value by adding random values r_{SNR_i}, each $\ell_{SNR_f} + \kappa$ bits long.

Note that the computation of the SNR of the whole signal can be obtained starting from the energy of the frames previously computed. In fact

$$SNR = \log_2 \frac{\sum_{j=1}^{mw}(fz_j)^2}{\sum_{j=1}^{mw}(fn_j)^2} = \log_2 \frac{\sum_{i=1}^{m}\sum_{j=1}^{w}(fz_j^i)^2}{\sum_{i=1}^{m}\sum_{j=1}^{w}(fn_j^i)^2}$$
$$= \log_2(\sum_{i=1}^{m} E_{fz^i}) - \log_2(\sum_{i=1}^{m} E_{fn^i}).$$

$\mathcal{C}$ adds together all the frame energies $E_{z,o'}$ and all the frame energies $E_{n,o'}$, obtaining the energies of the whole signals obfuscated by a value that is the sum of the obfuscation introduced in each frame energy, still known by $\mathcal{S}$. Finally, the SNR is obtained by another circuit similar to the one described above but having a larger input bit-length and hence a larger number of gates.

Once all the obfuscated SNR_i^f and SNR values are obtained, $\mathcal{C}$ encrypts them and transmits the ciphertexts to $\mathcal{S}$, who can remove the obfuscation by using homomorphic properties.

Mean and Variance computation. Obtained the values $[\![\mathbf{SNR^f}]\!] = \{[\![SNR_1^f]\!], \ldots, [\![SNR_m^f]\!]\}$, $\mathcal{S}$ can compute their mean and variance. The computation of the SNRs mean would require division by m. If m is public (or known), this is a cheap operation. However, if it is a private value, the division operation can be expensive and requires an interac-

tive protocol [22]. An additional disadvantage is that this protocol would introduce a rounding error. An alternative is to avoid the division by m. Then, the mean is amplified by a factor of m, the accuracy is preserved and the complexity of the protocol is reduced since interaction with $\mathcal{C}$ is not necessary. The amplified mean is simply computed by $\mathcal{S}$ as $[\![\mu^{SNR}]\!] = [\![\sum_{i=1}^{m} SNR_i^f]\!] = \prod_{i=1}^{m} [\![SNR_i^f]\!]$.

The computation of the variance can be performed by using an interactive protocol. Given the amplified mean, the SNR^f values have to be amplified by the same factor before computing the variance. Moreover, following the same approach we used for the mean, division by m is avoided and thus, the resulting variance is amplified by a factor m^3. Notice that $\mathcal{C}$ already has each SNR_i^f value obfuscated with a random value r_{SNR_i}. Thus, $\mathcal{S}$ can send the obfuscated SNR mean $[\![\mu^{SNR} + r_\mu]\!]$ (where $r_\mu \in \mathbb{Z}_{\ell_{\mu+\kappa}}$) to $\mathcal{C}$ who decrypts it and computes

$$\sum_{i=1}^{m} \left(m(SNR_i^f + r_{SNR_i}) - (\mu^{SNR} + r_\mu) \right)^2 .$$

Finally $\mathcal{C}$ encrypts the result and sends it back to $\mathcal{S}$ who removes the obfuscation value $\sum_i (m r_{SNR_i} - r_\mu)^2 - 2\sum_i (m SNR_i^f - \mu^{SNR})(m r_{SNR_i} - r_\mu)$ by using homomorphic properties, thus, obtaining the encrypted amplified variance $[\![\sigma^{SNR}]\!]$.

Classification. The previously computed values (the overall SNR, the frame SNR mean and the frame SNR variance) can be used to develop simple classifiers that compare them with thresholds obtained by training. In the following, we show that by combining them, one can improve the accuracy of the whole classification by using a simple linear classifier. To do so, we developed a linear classifier that uses σ^{SNR}, μ^{SNR} and the SNR computed on the whole signal as input. In practice the signal quality is classified by evaluating the following inequality:

$$a + b\ \sigma^{SNR} + c\ \mu^{SNR} + d\ SNR > 0$$

where the coefficients a, b, c, d are obtained by training. If the inequality is true the signal is classified as noisy, otherwise as clean.

Training is made in the plain domain independently for each subject by using ECG recorded in a supervised environment so that the weight vector $\boldsymbol{\beta} = [a, b, c, d]$ can be obtained. Assume that a training set of k segments has been recorded from a patient and that from each segment j, a vector $\alpha_{\mathbf{j}} = [1, \sigma_j^{SNR}, \mu_j^{SNR}, SNR_i]$ is created in the plain domain, including an additional bit γ_j, indicating whether the signal has been classified as clean ($\gamma_j = -1$), or noisy ($\gamma_j = +1$). Then, we obtain,

$$a + b\ \sigma_j^{SNR} + c\ \mu_j^{SNR} + d\ SNR_j + \epsilon_j = \boldsymbol{\alpha}_j\ \boldsymbol{\beta} + \epsilon_j = \gamma_j$$

where ϵ_j is the error committed by the classifier.

By considering all the signals available for training, we compute the matrix $\mathbf{A} = [\alpha_1, \ldots, \alpha_k]^T$ and the vector $\mathbf{G} = [\gamma_1, \ldots, \gamma_k]^T$. The goal of the training is finding the vector $\boldsymbol{\beta}$ that minimizes the mean square error sum $\sum_{i=1}^{k} \epsilon_i^2$, which can be obtained as

$$\boldsymbol{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{G}.$$

Because $[a, b, c, d]$ are real values they are quantized and represented with integer numbers to be used in the encrypted domain. Observe that the scalar product can be implemented by resorting to HE only[2]. It is important to point out that the number of bits required to represent the data depends on many factors and it can change with each training set. In particular, the bitsize for the coefficients can be chosen so that the computation precision is similar to that of a plain implementation. In short, the classification is computed using HE:

$$[\![a + b\ \sigma^{SNR} + c\ \mu^{SNR} + d\ SNR]\!] = \\ = [\![a]\!]\ [\![\sigma^{SNR}]\!]^b\ [\![\mu^{SNR}]\!]^c\ [\![SNR]\!]^d.$$

The sign of the scalar product is used for the classification, while the magnitude gives an indication of its reliability. High values are more reliable than smaller ones. Noticing that both sign and magnitude are useful for the classification and depending on the privacy requirements of $\mathcal{S}$ and $\mathcal{C}$, the protocol can choose to disclose the scalar product result to $\mathcal{C}$. Hence, $\mathcal{S}$ sends the encrypted magnitude and sign values to $\mathcal{C}$ who decrypts them and performs the final classification (determines to which class the particular measurement belongs). If $\mathcal{S}$ prefers to kept the magnitude secret, it can obfuscate the scalar product result by using a random value and then transmit a comparison GC having the obfuscated value and the random value as inputs.

4. ANALYSIS

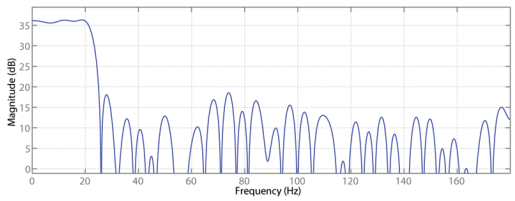
In a real implementation the system has to be trained on patient's data in a supervised environment. A nurse or a doctor prepares a training set with intervals of clean ECG obtained with electrodes correctly and wrongly placed by them and by the patient. To increase the data set, simulated noise can be added with different power levels to the whole clean signals or only in small random intervals. Finally, the nurse subdivides the ECG in clean and noisy segments. The subdivision can be made as a function of subjective parameters (an expert decides if the quality is sufficient or not) or testing the segments with the software that will be later used for the analysis of the ECG, when the software returns the answer expected by the expert, the signal is classified as clean, otherwise as noisy. The obtained data set is used to train a classifier that is then used to evaluate the signals that the patient will record at home connecting the electrodes without the presence of an expert.

We used the MIT-BIH Arrhythmia Database[3] for our experiments. The signals were subdivided into intervals and classified as noisy or clean according to the annotation available in the database, even if the type of noise was not specified explicitly. An interval is considered clean if no samples are affected by noise, otherwise it is considered clean. To extend the data set we added artificial electrode contact noise stored in the MIT-BIH Noise Stress Test Database[4] to whole clean

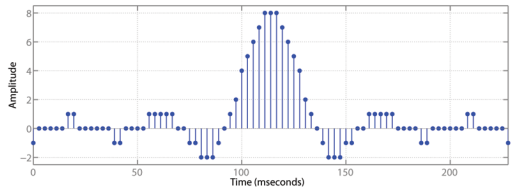
[2]This assumes that the coefficients are known in plaintext form to $\mathcal{S}$. Alternatively, a two party multiplication protocol can be performed similar to the computation of the variance. For reasons of brevity, this protocol is not included in this version of the paper.

[3]http://www.physionet.org/physiobank/database/mitdb/

[4]http://www.physionet.org/physiobank/database/nstdb/



(a) Spectrum of the filter(Magnitude response)



(b) Impulse response

Figure 2: Filter plots

segments and partially (only to a randomly chosen section of the segment).

4.1 Data Dimension

To evaluate the quality of the signal we propose to analyze $\tau = 30$ seconds of the signal. This interval length is chosen because it is long enough to allow a correct evaluation and not so long to delay the beginning of the subsequent analysis, especially if the patient places the electrodes in a wrong way and has to evaluate the quality of the signal several times.

The signals in the MIT-BIH Arrhythmia database have a sample frequency $f_s = 360$Hz and are recorded in the U.S.A. (with a power-line frequency $f_p = 60$Hz). Each sample is represented with $\ell_x = 10$ bits for the magnitude and one bit for the sign, hence the maximum value it can assume is $max_x = 1023$. The same characteristics are considered for y, the signal obtained after filtering with the protocol proposed by Van Alsté and Schidler [21] to remove the baseline wonder and the power-line noise, hence $\ell_y = 10$ and $max_y = 1023$.

The filter is an integer low pass filter having sample frequency 360Hz and cut-off frequency 20Hz and it is considered to be owned by $\mathcal{S}$ (i.e., its coefficients should remain private to $\mathcal{S}$). It was obtained by quantizing a filter developed in Matlab with *fdatool*. Figure 2 shows the spectrum of the filter, and its impulse response (the coefficients). The filter has been optimized to obtain a small number of coefficients that can be represented with a few bits. The filter has order 82, hence it can be represented with 83 coefficients (only 42 coefficients are needed thanks to symmetry) 36 of which are zero and the remaining ones assume values between -2 and 8. Hence, each coefficient can be represented with $\ell_c = 4$ bits for the magnitude and 1 for the sign. The amplification introduced by the integer filter is $amp = 64$, while the sum of the coefficients is $|c_0| + \sum_{j=1}^{k} |2 * c_j| = 112$. This sum is used to estimate the maximum value a filtered sample can assume.

In the following, we provide an analysis of the bit-length of the data involved in the computation by using a worst case analysis. The maximum value that each variable can assume can be easily determined by a logarithm computation. Since the maximum value that a sample can assume is $max_y = 2^{\ell_y} - 1$, it follows that the maximum value that a filtered sample can assume is $max_z = |c_0 * max_y| + \sum_{j=1}^{k} |c_j * 2 * max_y| = max_y * (|c_0| + \sum_{j=1}^{k} |2 * c_j|)$, and its representation needs ℓ_z bits for the magnitude and 1 for the sign. The maximum value that a noise sample can assume is $max_z - (-ampmax_y)$. The magnitude of each noise sample can be represented with ℓ_n bits and another bit is required for the sign.

After some experiments we decided to subdivide the 30 seconds of signal ($t * f_s = 10800$ samples) into $m = 30$ frames, each having length one second ($w = 360$ samples). In the worst case, all filtered signal (or noise) samples assume the maximum value. The maximum value of the energy of a frame of the filtered signal is $max_{E_z} = \sum_{i=1}^{w} max_z^2 = w * max_z^2$ and the maximum energy of the noise frames is $max_{E_n} = wmax_n^2$. For simplicity, we represent both max_{E_z} and max_{E_n} with the same number of bits ℓ_e, obtained by the logarithm of max_{E_n}. Since the energy is positive, it is not necessary to add a bit for the sign. Similarly, the maximum value that both the energies of the whole filtered signal and noise signal can attain is $mw(max_z + ampmax_y)^2$, which can be represented with ℓ_E bits.

By using max_{E_n} as an upperbound for both the filtered and noise signal energies, the highest magnitude of the SNR is obtained when the energy of one of them is maximum and the other is zero. Note that this scenario is not possible, in practice. Thus, the maximum value that the frame SNR can assume is $max_{SNR_f} = \lceil \log_2(max_{E_{fz^i}}) \rceil = \pm \lceil \log_2(w * (2^{\ell_s} - 1)^2 * (|c_0| + \sum_{j=1}^{k} |2 * c_j|^2) \rceil$ and it can be represented with $\ell_{SNR_f} = \lceil \log_2 \ell_e \rceil$ bits for the magnitude and one bit for the sign. The SNR of the whole signal needs $\ell_{SNR} = \lceil \log_2 \ell_E \rceil + 1$ bits for its representation.

The highest value that the mean of the SNR can assume is $m * max_{SNR_f}$ and needs ℓ_μ bits for its magnitude representation and one for the sign. The differences between the SNR values amplified by m and the mean require another bit.

Finally we can obtain the maximum value for the variance by considering that

$$\begin{aligned} \sigma^{SNR} &= \sum_{j=1}^{m} (mSNR_{f\,j} - \mu^{SNR})^2 \\ &< \sum_{j=1}^{m} (mmax_{SNR_f} + mmax_{SNR_f})^2 \\ &= m^3 max_{SNR_f}^2 = max_\sigma \end{aligned}$$

and needs ℓ_σ bits for its representation. Being the variance positive, the sign bit is not needed.

We do not provide limitation for the representation of the classification parameters because their values depend on the training set and different bit-lengths can be necessary for different people. The correct bit-length is hence chosen so that classification with quantized parameters is sufficiently

Table 1: Maximum value and number of bits necessary for the magnitude representation of the variables involved in the computation by worst case analysis. Another bit is needed for the sign, except for energies and SNR variance.

Variable Name	Maximum Value	Magnitude Bitlength
Original Signal x	1023	$\ell_x = 10$
Pre-filtered Signal y	1023	$\ell_y = 10$
Filter Coefficients c	8	$\ell_c = 4$
Filtered Signal z	114576	$\ell_z = 17$
Noise Signal n	180048	$\ell_n = 18$
Frame Energy E_{fz}, E_{fn}	11670221629440	$\ell_e = 44$
Frame SNR SNR_f	44	$\ell_{SNR_f} = 6$
Full Energy E_z, E_n	350106648883200	$\ell_E = 49$
SNR SNR	49	$\ell_{SNR} = 6$
SNR Mean μ^{SNR}	1320	$\ell_\mu = 11$
SNR Variance σ^{SNR}	52272000	$\ell_\sigma = 26$

similar to the classification involving real parameters, paying attention that the scalar product result does not exceed the maximum value allowed by the ciphertext. In some cases, shortest bit-lengths will mean significant lower computational complexity. The results of the above analysis are summarized in Table 1.

4.2 Communication Complexity

This section provides an analysis of the communication complexity of the protocol (summarized in Table 2). For our analysis, we assume the use of the following parameters (short term security): $T = 1024$, $t = 80$, $\kappa = 80$ bits. The protocol starts with the transmission of $\tau f_s = 10800$ cyphertexts having size $2T$ from $\mathcal{C}$ to $\mathcal{S}$. During energy computation $\mathcal{S}$ transmits the filtered and noise samples to $\mathcal{C}$, together with the obfuscation values $r' - \sum_i 2r_i s_i$ that $\mathcal{C}$ has to remove from the energy. The $2\tau f_s$ samples can be packed in

$$\left\lceil \frac{2\tau f_s}{\lfloor T/(\ell_n + \kappa) \rfloor} \right\rceil$$

ciphertexts, while the $2m$ obfuscation values can be packed in $\lceil \frac{2m}{\lfloor T/(\max\{2\ell_n+\kappa+1, \ell_e+\kappa\}+1) \rfloor} \rceil$ciphertexts. At this point $\mathcal{C}$ needs to evaluate the GC. The circuit can be transmitted offline and is composed by m sub-circuits computing the frame SNR and one sub-circuit computing the whole SNR.

Since the energy values can be represented with ℓ_e bits, the GC that computes each SNR^f is composed by 2 subtraction circuits (ℓ_e non XOR gates), the circuit implementing the SNR computation $(2(\ell_e - 1) + \ell_e - \lceil \log_2(\ell_e + 1) \rceil + \ell_e = 4\ell_e - \lceil \log_2(\ell_e + 1) \rceil - 2$ non-XOR gates) and a controlled addition/subtraction circuit [10] that blinds the result ($\ell_{SNR_f} + \kappa$ non-XOR gates). Similarly the SNR from the whole filtered and noise signal energies (having bitlength ℓ_E) is computed and the circuit related is composed by $6\ell_E - \lceil \log_2(\ell_E + 1) \rceil - 2 + \ell_{SNR} + \kappa$ non-XOR gates. The whole circuit is hence composed by $m(6\ell_e - \lceil \log_2(\ell_e + 1) \rceil - 2 + \ell_{SNR_f} + \kappa) + 6\ell_E - \lceil \log_2(\ell_E + 1) \rceil - 2 + \ell_{SNR} + \kappa$ non-XOR gates having size $4t$ bits each. $\mathcal{C}$'s inputs to GC are $2m$ energies represented with ℓ_e bits each and 2 energies represented by ℓ_E bits. The secrets relative to $\mathcal{C}$'s input are transmitted by using OT. Since the number of input bits is greater than $3t$, the complexity of the OT is reduced to the transmission of $\approx 6t^2 + 4(2m\ell_e + 2\ell_E)t$ bits offline and $2(2m\ell_e + 2\ell_E)t$ bits online. $\mathcal{S}$'s inputs to the GC are the secrets corresponding to the random values used to blind the energies at the input and the SNR at the output. Notice that the values can be generated offline. Thus, they can also be transmitted offline together with the circuit, resulting in the transmission of $2m\ell_e + 2\ell_E + m(\ell_{SNR_f} + \kappa) + (\ell_{SNR} + \kappa)$ secrets of size t bits. $\mathcal{C}$ sends $\mathcal{S}$ m ciphertexts containing the obfuscated frame SNR, one ciphertext containing the SNR and, finally, two ciphertexts to compute the SNR variance.

Table 2: Online and offline bandwidth (bits) required by the protocol

	Offline	Online
HE	0	26,626,048
circuit	3,402,240	0
$\mathcal{C}$ secrets to GC (OT)	914,560	438,080
$\mathcal{S}$ secrets to GC	432,320	0
Total	4,749,120	27,064,128

4.3 Classification Performance

From each signal in the dataset, we obtained segments that can be subdivided into clean (c) or noisy (n). A segment is considered noisy if classified as such in the Physiobank databases. In a real implementation an expert decides if the signals used for training are clean or noisy.

Due to the short length of the signals in the database (30 minutes), the number of segments for test is limited. Furthermore, if there are few (< 10) clean or noisy segments extracted from a Physiobank database signal, the signal is discarded by the test. We performed three different tests in which we trained and evaluated the classifiers on three different data sets. The first data set (c/n) was built by using only real clean and noisy sections. The second (c/a) data set was built by using clean sections and sections where artificial noise was added to whole clean sections (a), considering them as noisy sections. The last data set (c/p) is similar to the second one, but, instead of completely noisy sections, we used clean sections where the noise was added only to a randomly chosen interval (p).

For each signal 60% of the segments of the different types were randomly chosen for the training and the others were used for the testing. A different $\boldsymbol{\beta}$ vector was obtained for each individual (signal). Table 3 shows the results obtained by using the linear classifier proposed in Section 3. Moreover the table shows the performance that we can obtain by using only the mean of the frame SNR, the variance or the frame SNR of the whole section. The table contains the mean of the results of all the signals used for the tests. It is clear from these results that the combination of all derived values with a simple linear classifier significantly improves the classification results.

5. CONCLUSION

In this paper, we have proposed a protocol to evaluate the quality of an ECG signal specifically geared towards remote health monitoring applications. The algorithm that we propose divides the incoming signal into windows, computes the

Table 3: Performance of the protocol using the linear classifier or a single feature.

Test type	Linear classifier	σ^{SNR}	μ^{SNR}	SNR
c/n	0.8490	0.8158	0.7061	0.7325
c/a	0.8365	0.8005	0.8368	0.8234
c/p	0.7377	0.6729	0.6695	0.6666

signal-to-noise ratios of the windows and its first and second order moments (mean and variance). These values are then used together with the SNR of the whole signal to classify it as clean or noisy depending on the result of a linear classifier, whose training set is specific to each individual. Implementing such a classification method in the plain domain, we achieve more than 84% correct classification rate on signals of the MIT Arrhythmia database, while the classification accuracy obtained with ECGs where synthetic noise is added is a bit lower. We expect to be able to improve the quality of the classification result by using more advanced techniques (PCA,LDA, etc.). The privacy-preserving version would require the transmission of ≈ 3.4 Mbytes of data, making a real implementation feasible. We also describe how to map this methodology to the encrypted domain using both homomorphic encryption primitives and Yao's garbled circuits. Similarly to [4], the protocol complexity can be reduced by performing tests to determinate the real bitlength of the values involved.

6. REFERENCES

[1] U. Acharya, J. Suri, J. Spaan, and S. Krishnan. *Advances in cardiac signal processing*. Springer Verlang GmbH Berlin Heidelberg, 2007.

[2] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology – EUROCRYPT*, volume 2045 of *LNCS*, pages 119–135. Springer, 2001.

[3] M. Alfaouri, K. Daqrouq, I. Abu-Isbeih, E. Khalaf, A. Al-Qawasmi, and W. Al-Sawalmeh. Quality Evaluation of Reconstructed Biological Signals. *American Journal of Applied Sciences*, pages 187–193, 2009.

[4] M. Barni, P. Failla, R. Lazzeretti, A. Paus, A.-R. Sadeghi, T. Schneider, and V. Kolesnikov. Efficient privacy-preserving classification of ecg signals. *Information Forensics and Security, WIFS. IEEE International Workshop on*, pages 91 –95, 2009.

[5] M. Barni, J. Guajardo, and R. Lazzeretti. Privacy preserving evaluation of signal quality with application to ecg analysis. In *Information Forensics and Security (WIFS), IEEE International Workshop on*, pages 1–6. IEEE, 2010.

[6] D. Beaver. Precomputing oblivious transfer. In *Advances in Cryptology – CRYPTO*, volume 963 of *LNCS*, pages 97–109. Springer, 1995.

[7] T. Bianchi, A. Piva, and M. Barni. Composite Signal Representation for Fast and Storage-Efficient Processing of Encrypted Signals. *IEEE Transactions on Information Forensic and Security*, 5(1):180–187, 2010.

[8] J. Brickell, D. E. Porter, V. Shmatikov, and E. Witchel. Privacy-preserving remote diagnostics. In *ACM Conference on Computer and Communications Security (CCS)*, pages 498–507. ACM, 2007.

[9] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Public-Key Cryptography (PKC)*, LNCS, pages 119–136. Springer, 2001.

[10] P. Failla, M. Barni, R. Lazzeretti, A. Sadeghi, and T. Schneider. Privacy-preserving ecg classification with branching programs and neural networks. *Information Forensics and Security, IEEE Transactions on*, (99):1–1, 2011.

[11] J. Hansen and B. Pellom. An effective quality evaluation protocol for speech enhancement algorithms. In *Fifth International Conference on Spoken Language Processing*. Citeseer, 1998.

[12] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology – CRYPTO*, volume 2729 of *LNCS*. Springer, 2003.

[13] V. Kolesnikov and T. Schneider. Improved garbled circuit: Free XOR gates and applications. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5126 of *LNCS*, pages 486–498. Springer, 2008.

[14] R. Lazzeretti and M. Barni. Division between encrypted integers by means of garbled circuits. In *Information Forensics and Security (WIFS), IEEE International Workshop on*, pages 1 –6, 29 2011-dec. 2 2011.

[15] Q. Li, R. Mark, and G. Clifford. Robust heart rate estimation from multiple asynchronous noisy sources. *Physiological measurement*, 29:15–32, 2008.

[16] Y. Lindell and B. Pinkas. A proof of Yao's protocol for secure two-party computation. ECCC Report TR04-063, Electronic Colloq. on Comp. Complexity, 2004.

[17] H. Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In *Advances in Cryptology – ASIACRYPT*, volume 2894 of *LNCS*. Springer, 2003.

[18] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay — a secure two-party computation system. In *USENIX*, 2004. `http://www.cs.huji.ac.il/project/Fairplay`.

[19] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *ACM-SIAM Symposium On Discrete Algorithms (SODA)*, pages 448–457. Society for Industrial and Applied Mathematics, 2001.

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.

[21] J. Van Alste and T. Schilder. Removal of Base-Line Wander and Power-Line Interference from the ECG by an Efficient FIR Filter with a reduced Number of Taps. *IEEE Transactions on Biomedical Engineering*, 32:12, 1985.

[22] T. Veugen. Encrypted integer division. In *Information Forensics and Security (WIFS), IEEE International Workshop on*, pages 1–6. IEEE, 2010.

[23] A. C. Yao. How to generate and exchange secrets. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 162–167. IEEE, 1986.