EFFICIENT PROTOCOLS FOR SECURE ADAPTIVE FILTERING

Juan Ramón Troncoso-Pastoriza[1] *Fernando Pérez-González*[1,2,3]

1. **University of Vigo**, Signal Theory and Communications Department, SPAIN
2. **Gradiant** (Galician Research and Development Center in Advanced Telecommunications)
3. **University of New Mexico**, Electrical and Computer Engineering Department, Albuquerque, NM

ABSTRACT

The field of *Signal Processing in the Encrypted Domain* (SPED) has emerged in order to provide efficient and secure solutions for preserving privacy of signals that are processed by untrusted agents.

In this work, we study the privacy problem of adaptive filtering, one of the most important and ubiquitous blocks in signal processing nowadays. We examine several use cases along with their privacy characteristics, constraints and requirements, that differ in several aspects from those of the already tackled linear filtering and classification problems. Due to the impossibility of using a strategy based solely on current homomorphic encryption systems, we propose novel secure protocols for a privacy-preserving execution of the BLMS (Block Least Mean Squares) algorithm, combining different SPED techniques, and paying special attention to the trade-off between computational complexity, bandwidth, and the error produced due to finite-precision implementations.

***Index Terms*—** Privacy, Adaptive Filtering, Iterative Methods, Complexity, Error analysis.

1. INTRODUCTION

Signal Processing in the Encrypted Domain [1] (SPED) is an emergent research field that has arisen to effectively tackle the privacy problems involving signal processing, covering the multiple application scenarios where the need for privacy is clearly present, mainly those in which biological signals (fingerprints, faces, iris, DNA, ECG signals, MRI images,...) are involved, as they hold extremely sensitive information about users or patients. While the most efficient SPED primitives are those that perform linear fixed operations (like encrypted DCTs or linear filtering) through homomorphic encryption, most of the times Signal Processing needs to resort to adaptive filtering algorithms, due to their greater flexibility, higher responsiveness to changes in the environment, convergence to the optimal fixed solution in a stationary environment, and their optimality when the information about the signal characteristics is not complete, offering a much better performance than fixed filters. Current homomorphic cryptosystems cannot directly deal with adaptive filters due to cipher blowup after a given number of iterations (cf. [2]), as the scale factor used for quantizing the inputs before encryption increases after each homomorphic multiplication; on the other hand, full homomorphisms, like Gentry's [3], able of executing any circuit without the need of decryption, are still not practical, due to the huge size needed for the ciphertexts.

There have been notable contributions [1] involving the combination of garbled circuits (for non-linear operations) and homomorphic processing, and also private collaborative filtering [4], posed in a scenario where many parties must be involved, and thus, not applicable to our setting of secure adaptive filtering.

In this work, we present secure solutions for privacy-preserving adaptive filtering that involve homomorphic processing with packing strategies, garbled circuits and interactive protocols, for overcoming the limitations of the three technologies, while profiting from their respective advantages. We take the BLMS [5] (Block Least Mean Squares) algorithm as a prototypical example of a relatively simple but powerful and versatile adaptive filter, and we optimize the privacy solutions in terms of computation and communication complexity, and the effect of fixed-point arithmetic on the output error.

The used vectors will be represented by lower-case boldface letters. The encryption of a number x (components of vector $\boldsymbol{x}$) will be represented by $[\![x]\!]$ ($[\![\boldsymbol{x}]\!]$). The operations performed between encrypted and clear numbers will be indicated as if they were performed in the clear; e.g. $[\![\boldsymbol{X}]\!] \cdot \boldsymbol{b}$ will represent the encryption of $[\![\boldsymbol{X} \cdot \boldsymbol{b}]\!]$. Finally, communication complexity of each protocol will be denoted by C_{cm}, measured in bits.

The structure of this work is as follows: Section 2 presents some prototypical privacy scenarios for adaptive filtering; Section 3 sketches our privacy solutions; Section 4 gives the error analysis and the complexity results for them, and Section 5 concludes the paper.

2. PRIVACY SCENARIO AND TRUST MODEL

We will consider two parties, $\mathcal{A}$ and $\mathcal{B}$, both using an additively homomorphic cryptosystem in an asymmetric scenario, where $\mathcal{B}$ can only encrypt, but $\mathcal{A}$ possesses also the decryption key, and can perform both encryption and decryption.

The studied scenario of private filtering represents a problem of private data processing, in which one party $\mathcal{B}$ has clear-text access to the to-be-filtered sequence u_n, while the other party $\mathcal{A}$ will provide the reference sequence d_n; both parties' inputs must be concealed from each other. The system parameters can be known by both parties or be provided by one party; in our case, we assume that the update step μ is agreed by both parties. The outputs of the algorithm (the filtered signal y_n and the updated filter $\boldsymbol{w}_n$, with N_E coefficients) are provided in encrypted form, in order to be input to a subsequent private protocol.

Assuming semi-honest parties, our protocols can be proven statistically secure under the random oracle model through a simulator argument, due to the use of sequentially composed secure subblocks

This work was partially funded by Xunta de Galicia under projects "Consolidation of Research Units" 2010/85, SAFECLOUD (ref. 09TIC014CT), SCALLOPS (ref. 10PXIB322231PR) and VISAGE (ref. 10TIC008CT), by the Spanish Ministry of Science and Innovation under project COMONSENS (ref. CSD2008-00010) of the CONSOLIDER-INGENIO 2010 Program, by the Iberdrola Foundation through the Prince of Asturias Endowed Chair in Information Science and Related Technologies, and by the Spanish MEC FPU grant ref. AP2006-02580.

and the semantic security of the underlying cryptosystems. Some of the scenarios, mainly related to *multiuser communications*, where these private protocols can be applied, are[1]:

Private Interference Cancellation: The privacy of the involved signals' owners must be protected from the respective receivers.

Private Adaptive Beamforming: Privacy stems not only from the signals, but also from the spatial position of the transmitters.

Private Model-Reference Adaptive Control (MRAC): Used in many industrial contexts like robot manipulation, ship steering, aircraft control or metallurgical/chemical process control. Privacy in this setting involves the parameters of the controller and the behavior of the controlled system.

Current privacy-preserving solutions cannot be directly applied to these scenarios due to the cipher blowup problem. We present in the next section our novel solutions, that have a direct application in the aforementioned scenarios and yield efficient private protocols that overcome cipher blowup with an optimal trade-off between precision and complexity.

3. PROPOSED PROTOCOLS

In this section, we present different approaches in order to tackle the private implementation of the BLMS algorithm, and to overcome the limitations that the sole application of current homomorphic encryption (cipher blowup) and garbled circuits (high bandwidth and dependence on the representation bit-size) has in our scenario, while preserving an acceptable MSE. The cipher blowup problem is a serious drawback, as it limits the number of allowed iterations of the adaptive algorithm to $N_{\text{max iter}} = \left\lfloor \frac{n_{\text{cipher}}}{n_x + 3 \cdot n_f} \right\rfloor$, where n_x bits are used for representing each input, with n_f bits for the fractional part, and n_{cipher} being the bit size of the maximum representable number inside the cipher; $N_{\text{max iter}}$ is a few tens for typical values.

3.1. Hybrid Implementation

We have developed a hybrid protocol with a packing strategy for block processing (Alg. 1) that uses homomorphic computation for the bulk of the algorithm, and a quantization circuit to avoid blowup. Conversion protocols from homomorphic to binary representation and vice-versa are used to connect both parts of the protocol.

We can argue that the optimal balance between both parts in terms of efficiency is reached when applying quantization at every iteration, when the scaled output of the filter y'_k is obtained (cf. Alg. 1), using a quantization step of 2^{3n_f} to recover the initial precision of n_f fractional bits. This strategy achieves the minimum of communication complexity for the used garbled circuit (one quantization circuit per output sample), and also the minimum computation complexity (it keeps a constant scaling factor and avoids rescaling operations). A finer step is used for the quantization of filter coefficients ($3 \cdot n_f$ fractional bits), resulting in an improved behavior in terms of MSE (cf. Section 4).

The packing strategy [6] keeps a block of N_b input values into the same encryption, that must be subjected to the same joint processing. Hence, the filter must be kept constant for each group of packed samples, and we take advantage of the block structure of the BLMS algorithm [5], with update equation

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \mu \sum_{i=0}^{N_b - 1} \boldsymbol{u}_{n \cdot N_b + i} \cdot e_{n \cdot N_b + i}, \; e_k = d_k - y_k. \quad (1)$$

The usual choice of N_b for the Block LMS filter is $N_b = N_E$, as it yields the minimum computational complexity.

The packing factors 2^{n_b} are chosen to be powers of two; thus, the bit-conversion protocol automatically unpacks the numbers without any extra complexity, and the conversion to homomorphic encryption after the circuit evaluation is performed for each unpacked number in parallel.

Algorithm 1 Hybrid Block (HB) PrivateLMS Protocol

```
Inputs: $\mathcal{A}$: $\llbracket d_n \rrbracket$; $\mathcal{B}$: $u_n$,$\llbracket \boldsymbol{w}_0 \rrbracket$.
Outputs: $\llbracket y_n \rrbracket$.
  1. $\mathcal{B}$ packs the input vector as $X_j^{(k)} = \sum_{i=0}^{N_b-1} 2^{n_x+3n_f} \cdot u_{k \cdot N_b+i-j}, j = \{0, \ldots, N_E - 1\}$.
  2. $\mathcal{A}$ generates the first $m \leq N_{\text{iter}}$ garbled circuits for unpacking and
     parallel rescaling, and sends them to $\mathcal{B}$; the circuits for the remaining iterations
     can be generated and sent during the execution of the previous ones.
  3. for $k = 1$ to $\lceil N_{\text{iter}} / N_b \rceil$
       (a) $\mathcal{B}$ multiplies $\llbracket \boldsymbol{y}'_k \rrbracket = \llbracket \boldsymbol{w}_k \rrbracket \cdot \boldsymbol{X}^{(k)}$, and apply the bit conversion protocol to $\llbracket \boldsymbol{y}'_k \rrbracket$.
       (b) $\mathcal{B}$ gets, through oblivious transfer ($OT$), the input keys to initialize the circuit
           corresponding to the $k^{\text{th}}$ iteration and executes it.
       (c) The resulting $\llbracket y_{k \cdot N_b+i} \rrbracket_b, i = \{0, \ldots, N_b - 1\}$ is converted back to a homomorphic
           encryption $\llbracket y_{k \cdot N_b+i} \rrbracket, i = \{0, \ldots, N_b - 1\}$, and then outputted.
       (d) $\mathcal{B}$ obtains $\llbracket e'_{k \cdot N_b+i} \rrbracket = \mu \cdot (\llbracket d_{k \cdot N_b+i} \rrbracket - \llbracket y_{k \cdot N_b+i} \rrbracket), i = \{0, \ldots, N_b - 1\}$.
       (e) $\mathcal{B}$ multiplies $\llbracket \boldsymbol{\Delta w}_k \rrbracket = \sum_{i=k \cdot N_b}^{(k+1) \cdot N_b - 1} \llbracket e_i \rrbracket \cdot \boldsymbol{u}_{i-N_E+1}$.
       (f) $\mathcal{B}$ updates the coefficients vector $\llbracket \boldsymbol{w}_{k+1} \rrbracket = \llbracket \boldsymbol{w}_k \rrbracket + \llbracket \boldsymbol{\Delta w}_k \rrbracket$.
```

3.2. Fast Implementation

The hybrid block protocol is far more efficient than using only garbled circuits, but the conversion protocols introduce an overhead, and the fact that the input values to the rounding garbled circuits are generated on the fly prevents much of the preprocessing that garbled circuits would need to compensate the complexity of the oblivious transfers. The gap in computational complexity with respect to only using homomorphic processing is too big (cf. Section 4.2), especially when using a high precision bit representation. In order to tighten that gap, we can substitute the circuits by an approximate rounding protocol with statistical security (Alg. 2). It can be seen that the rounding error that it introduces is higher than that of a linear quantizer, and it is not uniform between $[-\frac{1}{2}, \frac{1}{2})$, but triangular between $[-1, 1)$, thus duplicating the quantization MSE.

The allowed number of packed coefficients is reduced to $N_b^{(FB)} \leq \lfloor \frac{n_{\text{cipher}}}{n_b + n_{\text{sec}}} \rfloor$, instead of $N_b^{(HB)} \leq \lfloor \frac{n_{\text{cipher}} - n_{\text{sec}}}{n_b} \rfloor$, where $n_b = n_x + 3n_f$ is the maximum number of bits that a coefficient can occupy, and n_{sec} is the number of security bits required for the protocol. In this case, the approximate rounding protocol also performs the unpacking of the results. The implementation of this fast protocol substitutes the generation and use of the garbled circuits in the hybrid protocol by the much more efficient approximate rounding protocol. The disadvantage is that the rounding error rises; however, this is by far compensated by a reduction of the complexity gap with respect to the solely homomorphic solution.

[1] Further details of the application of our protocols to these scenarios can be found in [2]; we omit them here due to space constraints.

Algorithm 2 Approximate Rounding and unpacking Protocol

```
Inputs: $\mathcal{A}$: Quantization step $\Delta = 2^l$ and a security parameter $n_{sec}$;
$\mathcal{B}$: $[\![x_{\text{pack}}]\!] = [\![\sum_{i=0}^{N_b-1} x_i \cdot 2^{i\cdot(n_b+n_{sec}+1)}]\!]$, $\Delta = 2^l$, $n_{sec}$
Outputs: $\{[\![Q'_\Delta(x_i)]\!]\}_{i=0}^{N_b-1}$.
  1. $\mathcal{B}$ generates $x_i^{(b)} \in_R \{2^{n_b-1}, \dots, 2^{n_b-1} + 2^{n_b+n_{sec}}\}, i = \{0, \dots, N_b - 1\}$, with which he homomorphically shifts and additively blinds the packed encryptions: $[\![x_p^{(a)}]\!] = [\![x_{\text{pack}}]\!] + [\![\sum_{i=0}^{N_b-1} x_i^{(b)} \cdot 2^{i\cdot(n_b+n_{sec}+1)}]\!]$.
  2. $\mathcal{A}$ decrypts and unpacks $\{x_i^{(a)}\}_{i=0}^{N_b-1}$.
  3. Both parties apply a linear quantizer with step $\Delta = 2^l$ to their clear-text vectors component-wise, obtaining $\{Q_\Delta(x_i^{(a)})\}_{i=0}^{N_b-1}$ and $\{Q_\Delta(x_i^{(b)})\}_{i=0}^{N_b-1}$, respectively.
  4. $\mathcal{A}$ encrypts her quantized vector and sends the encryptions back to $\mathcal{B}$.
  5. $\mathcal{B}$ homomorphically unblinds the quantized encrypted values obtained from $\mathcal{A}$, obtaining the encrypted quantizations of the original values $\{[\![Q'_\Delta(x_i)]\!]\}_{i=0}^{N_b-1} = \{[\![Q_\Delta(x_i^{(a)})]\!] - Q_\Delta(x_i^{(b)})\}_{i=0}^{N_b-1}$.
```

4. FINITE PRECISION EFFECTS AND EVALUATION

In this section, we compare the developed protocols in terms of bandwidth, computational complexity and finite precision effects, providing also an evaluation of a practical implementation of our protocols, for measuring actual execution times on real machines.

One inherent limitation to privacy-preserving techniques dealing with finite-field based encryption is the need of using fixed-point arithmetic. Hence, numerical stability and numerical accuracy of the filters come into play. While this issue is commonly avoided or mitigated by the use of a sufficiently large plaintext size to accommodate the needed precision, it is necessary to predict which is the required precision and the needed plaintext size for keeping the output MSE within a given bound. We extend the error analysis of adaptive algorithms working with fixed-point arithmetic and apply it to our protocols. We assume that the inputs and outputs are quantized with n_f bits for their fractional part (of the total n_x bits for coding), and the filter coefficients and some intermediate results are quantized with n_{wf} bits and n_{If} bits for their fractional part respectively.

Neglecting the overflow effects and assuming stationary d_n and u_n [7] with variances σ_d^2 and σ_u^2, i.i.d.[2] u_n, and uniform and independent quantization errors of the inputs (with variance $\sigma^2 = \frac{2^{-2n_f}}{12}$) and intermediate values (with variance $\sigma_I^2 = \frac{2^{-2n_{If}}}{12}$, and $\sigma_w^2 = \frac{2^{-2n_{wf}}}{12}$ for the filter coefficients), it can be shown that the output average MSE in steady-state for the BLMS algorithm is [2]

$$\begin{aligned}\sigma_o^2(c,d,N_b) =& \sigma_{\text{min}}^2 + \frac{\mu\sigma_{\text{min}}^2 \text{tr}\boldsymbol{R}}{2-\mu\text{tr}\boldsymbol{R}} + \left(||\boldsymbol{w}^*||^2 + \frac{1}{2}\mu\sigma_{\text{min}}^2 N_E\right)\sigma^2 + c\sigma_I^2 \\ &+ \frac{\frac{N_E\sigma_w^2}{N_b} + d\cdot\left(N_E\frac{N_b-1}{N_b}\sigma_w^2 + \sigma_I^2\text{tr}(\boldsymbol{R})\right)}{2\mu - \mu^2 N_b \text{tr}\boldsymbol{R}} \\ &+ \frac{\mu^2\cdot\sigma^2\left(\left(1 + c\frac{\sigma_I^2}{\sigma^2} + ||\boldsymbol{w}^*||^2\right)\cdot\text{tr}\boldsymbol{R} + \sigma_{\text{min}}^2 N_E\right)}{2\mu - \mu^2 N_b\text{tr}\boldsymbol{R}}, \quad (2)\end{aligned}$$

where the first two terms correspond to the error of the (B)LMS filter with infinite precision, and the remaining terms stem from quantization. In Eq. (2), $\sigma_{\text{min}}^2 = \sigma_d^2 - \boldsymbol{w}^* E\{d_n\mathbf{u}_n\}$ is the error of the optimum Wiener filter $\boldsymbol{w}^*$, tr$\boldsymbol{R}$ is the trace of the input covariance matrix, N_b is the block size and c and d are factors reflecting the way quantization is handled in multiplications: $c = 1$ if only the result of $y_n = \boldsymbol{w}_n^T \cdot \boldsymbol{u}_n$ is quantized, and $c = N_E$ when quantization follows each intermediate product, while $d = 1$ when each product in $\mu\sum_k e_k\boldsymbol{u}_k$ is individually quantized, and $d = 0$ otherwise.

It can be seen that, when $d = 0$, BLMS reduces the sensitivity to the quantization error in the filter coefficients, that has a much more critical and noticeable effect than quantization of the input values when σ^2 and σ_w^2 are comparable, being BLMS much better behaved than LMS; nevertheless, when $\sigma^2 \gg \sigma_w^2$, BLMS and LMS produce a similar MSE. The hybrid block protocol presents an error at the output given by $\sigma_{\text{HB}}^2 = \sigma_o^2(1, 0, N_b)$, while a protocol based solely on garbled circuits would yield $\sigma_{\text{GC}}^2 = \sigma_o^2(N_E, 1, 1)$. For the fast protocol, the quantization error has a different shape, but the independence assumptions can be applied exactly as in the other protocols, with $\sigma_I^2 = \frac{2^{-2n_{If}}}{6}$.

Fig. 1 shows a representative case of the excess MSE (i.e., $E\{e^2\} - \sigma_{LMS_\infty}^2$) w.r.t. the infinite precision BLMS, obtained for varying bit-size of the fractional part. The theoretical approximations given by Eq (2) are labeled with the subindex th, and the experimental results[3], with exp. The hybrid protocol (HB for $N_b = N_E$ and Hy for $N_b = 1$) presents a much lower MSE than a protocol based solely on Garbled Circuits (GC), due to the use of a higher resolution for the vector coefficients, and due to quantizing only outputs, while the fast protocol (FB and FP) presents a MSE slightly higher than the hybrid protocol, due to the approximate quantization of the outputs. The block versions do not produce a noticeable impact on the MSE. The concordance between the theoretical approximation and the experimental results in all the protocols is remarkable, given the magnitude of the errors with which we are working, assessing the validity of Eq (2).

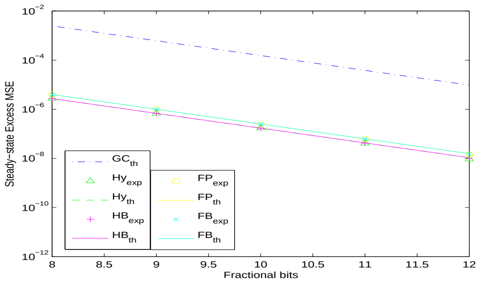

Fig. 1: Steady-state excess error for varying fractional precision, with $n_x = 48$ and $N_b = N_E = 12$ for the block protocols.

The value of N_b is limited by the maximum plaintext size and the number of bits used for representing each number. Thus, Eq. (2) can be used together with the packing limits for the block protocols $N_b^{(FB)} \leq \lfloor\frac{n_{\text{cipher}}}{n_b+n_{\text{sec}}}\rfloor$, $N_b^{(HB)} \leq \lfloor\frac{n_{\text{cipher}}-n_{\text{sec}}}{n_b}\rfloor$, for finding a trade-off between the committed error due to the used precision, and the complexity of both protocols, dependent on the number of coefficients that are packed together.

[2] The calculations can be generalized to any u_n through the rotated or uncoupled coordinate space, but the i.i.d. case is representative enough of the effects of fixed-point precision on the output error.

[3] Obtained as the average error for 40968 iterations in steady-state regime, for the system identification setup with $\sigma_u^2 = 0.25$, $\sigma_d^2 = 0.2821$, $\mu = 2^{-8}$, $\sigma_{\text{min}}^2 = 2.5\cdot 10^{-5}$ and $\sigma_{LMS_\infty}^2 = 2.5147\cdot 10^{-5}$. A protocol based solely on homomorphic processing would experiment cipher blow up before reaching the steady-state.

4.1. Bandwidth

In terms of bandwidth, the two developed protocols present the following communication complexity, (with XOR gates free of communication for the used implementation):

$$\begin{aligned}\mathrm{C}_{HBcm} =& \mathrm{C}_{Hycm} = (N_E - 1 + 3N_{\text{iter}} + 5N_E N_{\text{iter}})|E_H| \\ & + N_{\text{iter}}|E_C|(19n_x + 7n_{\text{sec}} + 24n_f), \\ \mathrm{C}_{FBcm} =& \left(\left(3 + \frac{1}{N_b}\right) N_{\text{iter}} + N_E - 1\right) |E_H|.\end{aligned}$$

Both have a complexity linear in the number of iterations, size of the filter and size of the encryptions; while the hybrid protocol's complexity is linear on the bit size of the numbers and independent of the number of packed coefficients, the fast one presents a significantly lower overhead, of the same order as using only homomorphic processing. Fig. 2 shows the number of communicated bits for each of the protocols (including a homomorphic processing protocol HP with no requantization, as a reference) when varying the filter length for a fixed number of iterations. The bandwidth using only garbled circuits (GC) is orders of magnitude higher than that of the fast solutions, while the hybrid protocol yields an intermediate complexity.

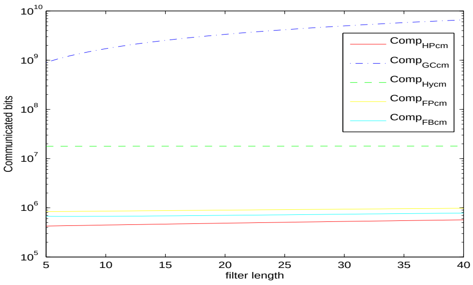

Fig. 2: Communication complexity as a function of the filter length with 50 iterations for $|E_H| = 4096$, $|E_C| = 224$, $n_{\text{sec}} = 80$, $N_b = \min\left(N_E, \lfloor \frac{n_{\text{cipher}}}{n_x + 3 \cdot n_f + n_{\text{sec}}} \rfloor\right)$, $n_x = 32$, $n_f = 16$.

4.2. Computational Load

In order to evaluate the computational complexity of the protocols, we have produced a C++ implementation using the Damgård-Jurik cryptosystem with some efficiency improvements in modular exponentiations, an XOR-free garbled circuit solution, efficient oblivious transfer (OT) protocols with EC-ElGamal encryptions (cf. [2] for details), aiming to the most efficient algorithms currently available for implementing garbled circuits. In order to measure only computation times, we have neglected the communication stack, and we have run in the same core the client and the server sequentially. Fig. 3 shows the resulting aggregated computation time[4] as a function of the filter size. The protocols involving garbled circuits are the most expensive ones, due to the load that OTs impose without precomputation. The packing performed in the block protocols allows for OT reductions, greatly improving computational load as $N_b = N_E$ increases. The execution times of the fast protocol are several orders of magnitude below those of the garbled circuits solutions, and slightly increase the complexity of a solely homomorphic computation protocol due to the addition of the rounding protocols. This is a remarkable result, as without this rounding subprotocols, the whole homomorphic computation protocol is completely unusable due to cipher blowup. For the fast protocol, the packing does not improve on the computational load, as it requires a whole unpacking protocol for each of the packed numbers, yielding the best efficiency for $N_b = 1$.

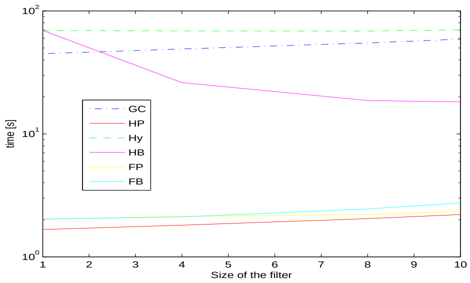

Fig. 3: Aggregated computation time for 2048 bits moduli, $|E_C| = 224$, $n_{\text{sec}} = 80$, $n_x = 32$, $n_f = 16$, 48 iterations and increasing filter size.

5. CONCLUSIONS AND FURTHER WORK

We have presented the problem of privacy-preserving adaptive filtering, with several representative scenarios. We have proposed several novel solutions for tackling the cipher blowup problem, employing different SPED techniques with an optimal trade-off in terms of complexity and precision; we have implemented all our novel protocols for the Private BLMS algorithm in a working prototype, evaluating it in terms of bandwidth and computational complexity and concluding that interactive approximate protocols with statistical security can yield much more practical solutions than garbled circuits.

We have analytically studied the effects of fixed-point precision on the output error in steady-state. Our fast protocols are almost as robust against quantization errors as the original (B)LMS algorithm, while presenting low complexity. This work opens the door to further research and improvements in secure adaptive filtering, setting the basis and a reference implementation for the development of new solutions.

6. REFERENCES

[1] "SPEED Project." http://www.speedproject.eu/

[2] J. R. Troncoso-Pastoriza and F. Pérez-González, "Secure adaptive filtering," *Submitted to IEEE TIFS*, 2010.

[3] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC'09*. Bethesda, MD, USA: ACM Press, May-June 2009, pp. 169–178.

[4] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Privacy-Preserving Centralized Recommender System," in *ACM SIGKDD*, 2010.

[5] S. Haykin, *Adaptive Filter Theory*, fourth edition ed., ser. Information and System Sciences. Prentice Hall, 2002.

[6] J. R. Troncoso-Pastoriza, S. Katzenbeisser, M. Celik, and A. Lemma, "A secure multidimensional point inclusion protocol," in *ACM MMSEC'07*, Sept. 2007, pp. 109–120.

[7] C. Caraiscos and B. Liu, "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 32, no. 1, pp. 34–41, Feb 1984.

[4] tested on an Intel Core2Duo processor at 3 GHz with 4GB of RAM running a 64-bit linux distribution.