# Secure Third-party Fast Covariance Intersection using Partially Homomorphic and Order Revealing Encryption Schemes

Marko Ristic, Benjamin Noack, and Uwe D. Hanebeck

***Abstract*— Fast covariance intersection is a widespread technique for state estimate fusion in sensor networks when cross-correlations are not known and fast computations are desired. The common requirement of sending estimates from one party to another during fusion means they do not remain private to their producing party. Current secure fusion algorithms have a reliance on encryption schemes that do not provide sufficient flexibility and as a result require, often undesired, excess communication between estimate producers. We propose a novel method of homomorphically computing the fast covariance intersection algorithm on estimates encrypted with a combination of encryption schemes. Using order revealing encryption we show how to approximate solutions to the fast covariance intersection coefficients can be computed and combined with partially homomorphic encryptions of estimates, to compute an encryption of the fused result. The described approach allows the secure fusion of any number of private estimates, making third-party cloud processing a viable option when working with sensitive state estimates, or when performing estimation over insecure networks.**

## I. Introduction

-
- Describe what will be in each following section and how the paper structure makes sense

### A. Notation

- Brief overview of ISAS vector and random vector notation
- Encryption notation, keys omitted where it's obvious from context
- Encryption of matrices and vectors is element-wise
- Real number encoding assumed in all encryption notation, all numbers are real

## II. Covariance Intersection and Approximations

- Covariance intersection (CI) is a method for fusing state estimations for different sources when models and cross-correlations are not known
- The fused estimate and estimate covariance are commuted by (1) and (2)
- Note that the estimate and estimate covariance depicted in (1) and (2) are in the Information filter form of the popular Kalman Filter as this simplifies computation.

$$\mathbf{P}^{-1} = \sum_{i=0}^{n} \omega_i \mathbf{P}_i^{-1} \tag{1}$$

$$\mathbf{P}^{-1}\underline{\hat{x}} = \sum_{i=0}^{n} \omega_i \mathbf{P}_i^{-1}\underline{\hat{x}}_i \tag{2}$$

$$\tag{3}$$

- Where values $\omega_i$ satisfy (4) and (5) and are chosen in a way to minimise a property of the resulting fused estimate.
- For example minimising the resulting trace would require solving (6)

$$\omega_0 + \omega_1 + \cdots + \omega_n = 1 \tag{4}$$

$$0 \leq \omega_i \leq 1 \tag{5}$$

$$\underset{\omega_0,\dots,\omega_n}{\arg\min}\{\operatorname{tr}(\mathbf{P})\} = \underset{\omega_0,\dots,\omega_n}{\arg\min}\{\operatorname{tr}((\sum_{i=0}^{n} \omega_i \mathbf{P}_i)^{-1})\} \tag{6}$$

- Minimising a given non linear cost function such as (6) can be very costly computationally and has led to the development of non-iterative approximation techniques [1], [2], [3]

### A. Fast Covariance intersection

- The fast covariance intersection (FCI) algorithm described in [1] is a common method used for approximating the solution to (6) by defining a new constraint on $\omega_i$ and solving that instead.
- In the 2 sensor case, (4) now becomes (7), and the additional requirement of (8) is also defined. Solutions are defined analytically and shown in (9)

$$\omega_0 + \omega_1 = 1 \tag{7}$$

$$\omega_0 \operatorname{tr}(\mathbf{P}_0) - \omega_1 \operatorname{tr}(\mathbf{P}_1) = 0 \tag{8}$$

$$\omega_0 = \frac{\operatorname{tr}(\mathbf{P}_1)}{\operatorname{tr}(\mathbf{P}_0) + \operatorname{tr}(\mathbf{P}_1)},\ \omega_1 = \frac{\operatorname{tr}(\mathbf{P}_0)}{\operatorname{tr}(\mathbf{P}_0) + \operatorname{tr}(\mathbf{P}_1)} \tag{9}$$

- When extending to any number of sensors, restriction (8) is generalised to (10)

$$\omega_i \operatorname{tr}(\mathbf{P}_i) - \omega_j \operatorname{tr}(\mathbf{P}_j) = 0,\ (i, j = 1, 2, \dots, n) \tag{10}$$

- Equation (10) is highly redundant and its largest linearly independent subset can be represented with (11).

$$\omega_i \operatorname{tr}(\mathbf{P}_i) - \omega_{i+1} \operatorname{tr}(\mathbf{P}_{i+1}) = 0,\ (i = 1, 2, \dots, n) \tag{11}$$

- The solution to (11) and (4) can be represented as the simultaneous equations problem shown in (12) where $\mathcal{P}_i = \operatorname{tr}(\mathbf{P}_i)$

Marko Ristic, Benjamin Noack, and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), Germany.
{marko.ristic,noack,uwe.hanebeck}@kit.edu

$$\begin{bmatrix} \mathcal{P}_0 & -\mathcal{P}_1 & 0 & \cdots & 0 \\ 0 & \mathcal{P}_1 & -\mathcal{P}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathcal{P}_{n-1} & -\mathcal{P}_n \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_0 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{12}$$

- Our goal for secure fusion is to solve (1), (2) and (12) using only encryptions from each sensor $i$.

## III. Homomorphic and Order Revealing Encryption

- To achieve a secure solution to the FCI fusion problem, we have focused on two types of function providing encryption schemes.
- Additive partially homomorphic encryption (PHE) schemes such ones defined in [4] and [5] provide a homomorphic addition operation as shown in (13).
- Order revealing encryption (ORE) schemes such as those in [6] and [7] provide a function which allows the comparison of encrypted values as shown in (14)

$$\mathcal{E}(a) \oplus \mathcal{E}(b) = \mathcal{E}(a+b) \tag{13}$$

$$f(\mathcal{E}(a), \mathcal{E}(b)) = cmp(a,b) \tag{14}$$

### A. Additive Partially Homomorphic Encryption

- We have used the Paillier encryption scheme described in [4] for the computation of the FCI due to its speed and implementation simplicity.
- The Paillier encryption scheme is a public key scheme, where encryptions are made with a public key but only the secret key holder can decrypt them.
- It provides two homomorphic operations on encrypted data, shown in (15) and (16). The modulus $N$ is computed as a product of 2 large primes which are part of the secret key.

$$\mathcal{E}(a)\mathcal{E}(b) \pmod{N} = \mathcal{E}(a+b \pmod{N}) \tag{15}$$

$$\mathcal{E}(a)^c \pmod{N} = \mathcal{E}(ca \pmod{N}),\ c \in \mathbb{Z}_N \tag{16}$$

- Encrypted numbers must be less than $N$, and negative numbers can be handled by storing integers in “two's complement” binary form, that is taking $[0, \frac{N}{2})$ as all possible positive numbers, and $[\frac{N}{2}, N)$ as the decreasing negative integers.

### B. Real Number Encoding for Homomorphic Encryption

- The Paillier encryption scheme can only encrypt, add, and multiply with integers. Due to the prevalence of real number values in sensor outputs and estimation processes, some form of encoding is required for these numbers to be encrypted.
- Real numbers, typically stored as floating-point numbers in sensor hardware, are converted to integers using the “Q” number format. A real number $a$ can be encoded to an integer $e$ using (17), where the largest encodable real number has an integer part of $i$ bits. Integer bits $i$ and fractional bits $f$ are chosen such that the largest encoded value can still be encrypted.

$$e = \begin{cases} \lfloor 2^f a \rfloor & a < 2^i \\ \lfloor 2^f (2^i - a) \rfloor & a \geq 2^i \end{cases} \tag{17}$$

- While the encoded real numbers are consistent under addition, multiplication by constants requires that a factor of $\frac{1}{2^f}$ be removed.
- Since division is not supported under the encryption scheme, the number of multiplications performed on an encrypted value must be bounded and handled when decoding. This also decreases the size of the largest encodable real number that can be decoded correctly.
- In our case, only a single multiplication is required, and decoding of an integer $e$ to a real number $f$ is performed by (18).

$$a = \begin{cases} \frac{e}{2^f} & e < 2^{(i+f)} \\ \frac{e}{2^{2f}} & e \geq 2^{(i+f)} \end{cases} \tag{18}$$

### C. Left-Right Order Revealing Encryption

- For the ORE scheme we have considered in particular the Left-Right encryption scheme described in [7] which will help in preventing information leakage as described in section IV.
- The key difference between this scheme and others is how numbers are compared. Left-Right encryption allows any number to be encrypted as either a “Left” or “Right” encryption, but only a “Left” encryption can be compared with a “Right” encryption.
- The ORE scheme described requires a single symmetric key for the encryption of either “Left” or “Right” encryptions, which can be compared without any key.

$$encrypt^L_{ORE}(sk, x) = \mathcal{E}^L_{ORE}(x) \tag{19}$$

$$encrypt^R_{ORE}(sk, y) = \mathcal{E}^R_{ORE}(y) \tag{20}$$

$$compare_{ORE}(\mathcal{E}^L_{ORE}(x),\ \mathcal{E}^R_{ORE}(y)) = cmp(x,y) \tag{21}$$

## IV. Secure Fast Covariance Intersection with 2 Sensors

$$\mathcal{E}(\mathbf{P}) = \mathcal{E}(\mathbf{P}_0)^{\omega_0}\mathcal{E}(\mathbf{P}_1)^{(1-\omega_0)} \tag{22}$$

$$\mathcal{E}(\mathbf{P}\underline{\hat{x}}) = \mathcal{E}(\mathbf{P}_0\underline{\hat{x}}_0)^{\omega_0}\mathcal{E}(\mathbf{P}_1\underline{\hat{x}}_1)^{(1-\omega_0)} \tag{23}$$

$$[\mathcal{E}^L_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_0^{-1})),\ \omega \in [0, 0+s, \ldots, 1-s, 1]] \tag{24}$$

$$\omega_0' = \frac{1}{2}(a+b),\ \omega_1' = (1-\omega_0) \tag{25}$$

## V. Multi-sensor Secure Fast Covariance Intersection

$$\omega_0\,\mathrm{tr}(\mathbf{P}_0) - \omega_1\,\mathrm{tr}(\mathbf{P}_1) = 0 \tag{26}$$

$$\omega_1\,\mathrm{tr}(\mathbf{P}_1) - \omega_2\,\mathrm{tr}(\mathbf{P}_2) = 0 \tag{27}$$

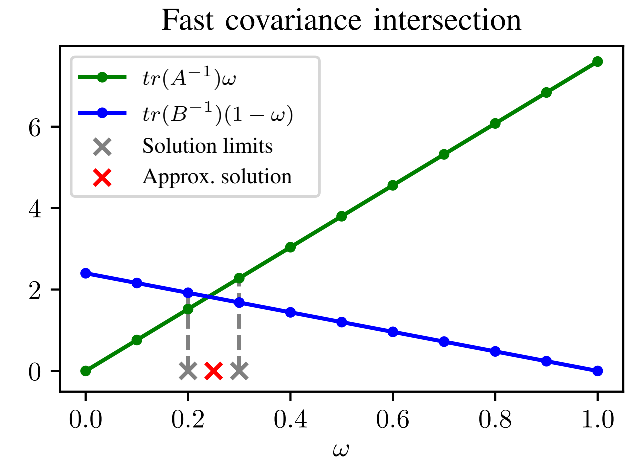

Fig. 1. Approximation of $\omega_0$ with discretisation step size $s = 0.1$. Only comparisons of the ordered values sent from either estimator are used.

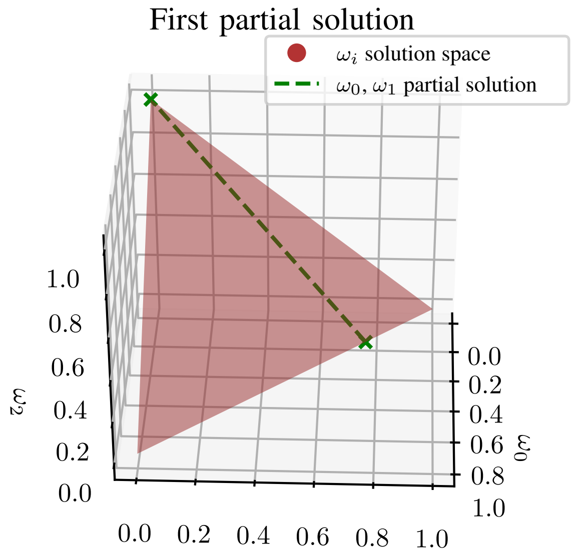

Fig. 2. Partial solution from equation (26) plotted on the plane of all possible values of $\omega_0$, $\omega_1$, and $\omega_2$.

$$a_0x + a_1y + a_2z + d = 0 \tag{28}$$

$$\begin{bmatrix} a_0^0 & a_1^0 & a_2^0 \\ a_0^1 & a_1^1 & a_2^1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} d^0 \\ d^1 \\ 1 \end{bmatrix} \tag{29}$$

$$\begin{bmatrix} a_0^0 & a_1^0 & \cdots & a_n^0 \\ a_0^1 & a_1^1 & \cdots & a_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_0^n & a_1^n & \cdots & a_n^{n-1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_{n-1} \\ \omega_n \end{bmatrix} = \begin{bmatrix} d^0 \\ d^1 \\ \vdots \\ d^{n-1} \\ 1 \end{bmatrix} \tag{30}$$

$$[\mathcal{E}^L_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_i^{-1})),\ \omega \in [0, 0+s, \dots, 1-s, 1]],\ i \text{ is even} \tag{31}$$

$$[\mathcal{E}^R_{ORE}(\omega\,\mathrm{tr}(\mathbf{P}_i^{-1})),\ \omega \in [0, 0+s, \dots, 1-s, 1]],\ i \text{ is odd} \tag{32}$$

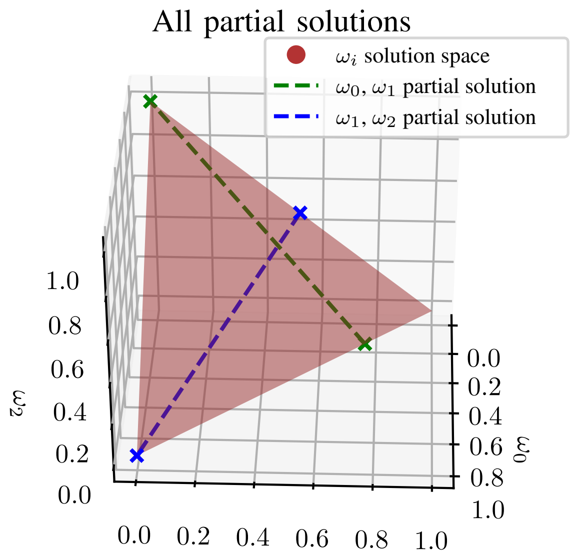

Fig. 3. Partial solutions from equations (26) and (27) plotted on the plane of all possible values of $\omega_0$, $\omega_1$, and $\omega_2$.

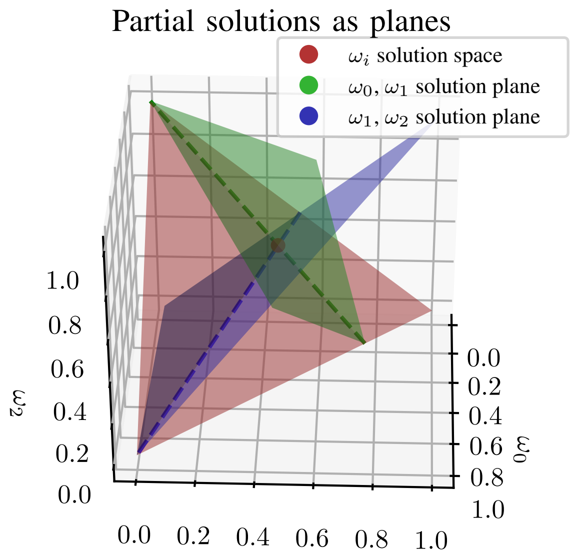

Fig. 4. Partial solutions from figure 3 plotted as planes perpendicular to the plane of possible solutions. Intersection point gives solution values of $\omega_i$ for Fast Covariance Intersection.

## VI. Simulation Results

## VII. Conclusion

## VIII. INTRODUCTION

Your goal is to simulate, as closely as possible, the usual appearance of typeset papers. This document provides an example of the desired layout and contains information

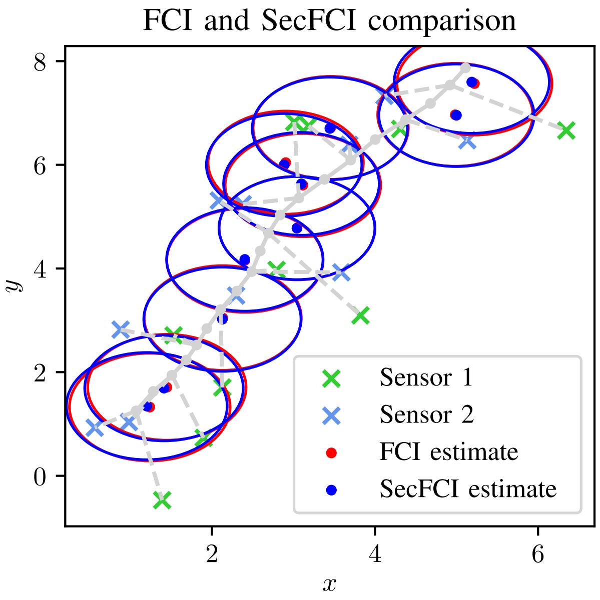

Fig. 5. Tracking simulation comparing Fast Covariance Intersection and our Secure Fast Covariance Intersection fusion methods.

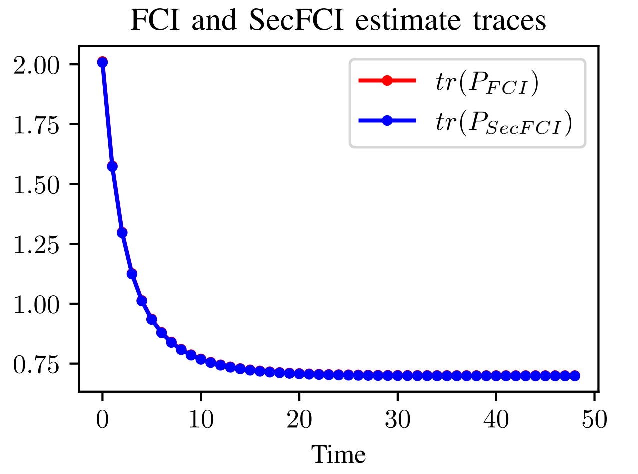

Fig. 6. Plot showing the fused estimate covariance trace throughout a tracking simulation, for both Fast Covariance Intersection and our Secure Fast Covariance Intersection

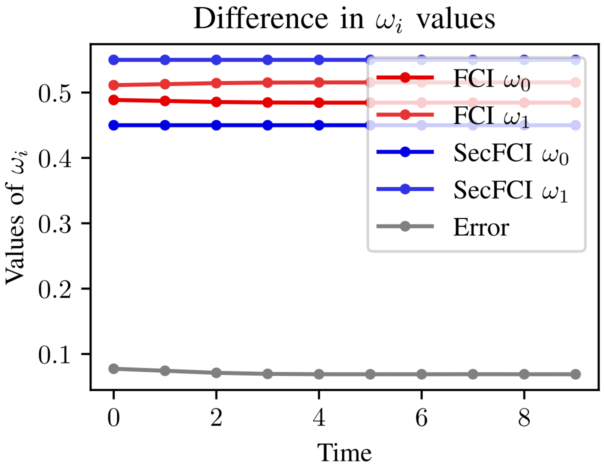

Fig. 7. Plot showing the difference in $\omega_i$ values between Fast Covariance Intersection and our Secure Fast Covariance Intersection, throughout a tracking simulation.

regarding desktop publishing format, type sizes, and type faces.

### A. *Full-Size Camera-Ready (CR) Copy*

If you have desktop publishing facilities, (the use of a computer to aid in the assembly of words and illustrations on pages) prepare your CR paper in full-size format, on paper 21.6 x 27.9 cm (8.5 x 11 in or 51 x 66 picas). It must be output on a printer (e.g., laser printer) having 300 dots/in, or better, resolution. Lesser quality printers, such as dot matrix printers, are not acceptable, as the manuscript will not reproduce the desired quality.

*1) Typefaces and Sizes::* There are many different typefaces and a large variety of fonts (a complete set of characters in the same typeface, style, and size). Please use a proportional serif typeface such as Times Roman, or Dutch. If these are not available to you, use the closest typeface you can. The minimum typesize for the body of the text is 10 point. The minimum size for applications like table captions, footnotes, and text subscripts is 8 point. As an aid in gauging type size, 1 point is about 0.35 mm (1/72in). Examples are as follows:

*2) Format::* In formatting your original 8.5” x 11” page, set top and bottom margins to 25 mm (1 in or 6 picas), and left and right margins to about 18 mm (0.7 in or 4 picas). The column width is 88 mm (3.5 in or 21 picas). The space between the two columns is 5 mm(0.2 in or 1 pica). Paragraph indentation is about 3.5 mm (0.14 in or 1 pica). Left- and right-justify your columns. Cut A4 papers to 28 cm. Use either one or two spaces between sections, and between text and tables or figures, to adjust the column length. On the last page of your paper, try to adjust the lengths of the two-columns so that they are the same. Use automatic hyphenation and check spelling. Either digitize or paste your figures.

## IX. UNITS

Metric units are preferred for use in IEEE publications in light of their international readership and the inherent convenience of these units in many fields. In particular, the use of the International System of Units (SI Units) is advocated. This system includes a subsystem the MKSA units, which are based on the meter, kilogram, second, and ampere. British units may be used as secondary units (in parenthesis). An exception is when British units are used as identifiers in trade, such as, 3.5 inch disk drive.

TABLE I

An Example of a Table

| | |
|---|---|
| One | Two |
| Three | Four |

## X. Additional Requirements

### A. Figures and Tables

Position figures and tables at the tops and bottoms of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table captions should be above the tables. Avoid placing figures and tables before their first mention in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence. Figure axis labels are often a source of confusion. Try to use words rather then symbols. As an example write the quantity "Inductance", or "Inductance L", not just. Put units in parentheses. Do not label axes only with units. In the example, write "Inductance (mH)", or "Inductance L (mH)", not just "mH". Do not label axes with the ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

### B. Numbering

Number footnotes separately in superscripts[1] Place the actual footnote at the bottom of the column in which it is cited. Do not put footnotes in the reference list. Use letters for table footnotes (see Table I).

### C. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title unless they are unavoidable.

### D. Equations

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). To make your equations more compact you may use the solidus (/), the exp. function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather then hyphen for a minus sign. Use parentheses to avoid ambiguities in the denominator. Punctuate equations with commas or periods when they are part of a sentence:

$$\Gamma_2 a^2 + \Gamma_3 a^3 + \Gamma_4 a^4 + ... = \lambda \Lambda(x),$$

where $\lambda$ is an auxiliary parameter.

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Use "(1)," not "Eq. (1)" or "Equation (1)," except at the beginning of a sentence: "Equation (1) is ...".

Fig. 8. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

## XI. Conclusions and Future Works

### A. Conclusions

This is a repeat. Position figures and tables at the tops and bottoms of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table captions should be above the tables. Avoid placing figures and tables before their first mention in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence. Figure axis labels are often a source of confusion. Try to use words rather then symbols. As an example write the quantity "Inductance", or "Inductance L", not just. Put units in parentheses. Do not label axes only with units. In the example, write "Inductance (mH)", or "Inductance L (mH)", not just "mH". Do not label axes with the ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

### B. Future Works

This is a repeat. Position figures and tables at the tops and bottoms of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table captions should be above the tables. Avoid placing figures and tables before their first mention in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence. Figure axis labels are often a source of confusion. Try to use words rather then symbols. As an example write the quantity "Inductance", or "Inductance L", not just. Put units in parentheses. Do not label axes only with units. In the example, write "Inductance (mH)", or "Inductance L (mH)", not just "mH". Do not label axes with the ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

## XII. Acknowledgments

The authors gratefully acknowledge the contribution of National Research Organization and reviewers' comments.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

[1] This is a footnote

# References

[1] W. Niehsen, “Information fusion based on fast covariance intersection filtering,” in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, vol. 2, pp. 901–904 vol.2, July 2002.

[2] D. Franken and A. Hupper, “Improved fast covariance intersection for distributed data fusion,” in *2005 7th International Conference on Information Fusion*, vol. 1, pp. 7 pp.–, July 2005.

[3] J. Cong, Y. Li, G. Qi, and A. Sheng, “An order insensitive sequential fast covariance intersection fusion algorithm,” *Information Sciences*, vol. 367-368, pp. 28–40, Nov. 2016.

[4] P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” in *Advances in Cryptology — EUROCRYPT ’99* (J. Stern, ed.), Lecture Notes in Computer Science, pp. 223–238, Springer Berlin Heidelberg, 1999.

[5] S. Goldwasser and S. Micali, “Probabilistic encryption,” *Journal of Computer and System Sciences*, vol. 28, pp. 270–299, Apr. 1984.

[6] N. Chenette, K. Lewi, S. A. Weis, and D. J. Wu, “Practical Order-Revealing Encryption with Limited Leakage,” in *Fast Software Encryption* (T. Peyrin, ed.), vol. 9783, pp. 474–493, Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[7] K. Lewi and D. J. Wu, “Order-Revealing Encryption: New Constructions, Applications, and Lower Bounds,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS’16*, (Vienna, Austria), pp. 1167–1178, ACM Press, 2016.