# Enhance Sensor Data Fusion Based on Time Slot Voting Mechanism

**Mr.A.Vigneshkumar[1], Mr.V.Balamurugan[2]**

[1] Assistant Professor, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Satyamangalam, Tamil Nadu, India
[2] Assistant Professor, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Satyamangalam, Tamil Nadu, India

***Abstract-*** A Virtual Sensor estimates product objective or process conditions using mathematical models rather than and sometimes in conjunction with physical sensors. These mathematical models use other physical sensor readings to calculate the estimated property or condition. Virtual sensor may provide flexibility, cost effective solutions, promote diversity, ensure security and increase manageability. In this work, data fusion is performed and privacy gets increase in virtual sensor. Data fusion is often performed in order to reduce the overall message transmission from the sensors toward the base station. Time slotted voting mechanism is used for data fusion. This work investigates the problem of data fusion assurance in multi-level data fusion or transmission in this paper. Different to a recent approach of direct voting where the base station polls other nodes directly regarding to the received fusion result, this work propose a scheme that uses the time-slotted voting technique. In this scheme, each fusion node broadcasts its fusion data or vote during its randomly assigned time slot. Only the fusion result with enough number of votes will be accepted. Thus, our scheme eliminates the polling process and eases the energy consumption burden on the base station or the fusion data receiver, which could well be the intermediate nodes. In this work, for security plumstead's algorithm is used to overcome against threat and attack

***Index Terms***- *Virtual sensor, Data fusion, Privacy*.

## I. INTRODUCTION

**Virtual sensor networks (VSNs)** is an emerging form of collaborative wireless sensor networks. In contrast to early wireless sensor networks that were dedicated to a specific application (e.g., target tracking), VSNs enable multi-purpose, collaborative, and resource efficient WSNs. The key idea difference of VSNs is the collaboration and resource sharing. By doing so nodes achieve application objectives in a more resource efficient way. These networks may further involve dynamically varying subset of sensor nodes.

A VSN can be formed by providing logical connectivity among collaborative sensors. Nodes can be grouped into different VSNs based on the phenomenon they track or the task they perform. VSNs are expected to provide the protocol support for formation, usage, adaptation, and maintenance of subset of sensors collaborating on a specific task(s). Even the nodes that do not sense the particular event/phenomenon could be part of a VSN as far as they are willing to allow sensing nodes to communicate through them. Thus, VSNs make use of intermediate nodes, networks, or other VSNs to efficiently deliver messages across members of a VSN.

It is necessary to incorporate appropriate secure mechanisms into virtual sensor networks. However, given the stringent constraints on processing power, memory, bandwidth, and energy consumption, it is very difficult to design suitable secure mechanisms for virtual sensor networks. This leave very limited resources for the necessary security components in VSNs. The constraints posed by the sensor hardware make it impossible to deploy most of the traditional security primitives and protocols. For example, it is too expensive to apply asymmetric cryptography to virtual sensor networks, such as the RSA and Diffie-Hellman algorithm, because they require expensive computations and long messages that could easily exhaust the sensor's resources.

## II. RELATED WORK

Projects targeted directly for sensor networks have often explored representing the sensor network as a database. Two demonstrative examples are TinyDB [17] and Cougar [18]. Generally these approaches enable applications with data requests that flow out from a central point (i.e., a base station) and create routing trees to funnel replies back to this root. These approaches focus on performing intelligent in-network aggregation and routing to reduce the overall energy cost while still keeping the semantic value of data high. In both approaches, data aggregation is specified using an SQL-like language. Queries cannot be used to merge different data types, i.e. only homogeneous data aggregation is possible. In contrast, the virtual sensors approach offers simple programming interface, supports multiple access points and offers raw and heterogeneous in-network data processing. A more lightweight implementation designed specifically for wireless sensor networks is TinyML [3]. It follows some of the SensorML ideas that are built on XML and has the important concept of virtualizing physical components.

VSN for distributed detection with N sensors for collecting environment variation data. The collected data are transmitted to a fusion node from all of the sensors. The fusion node yields a final result according to the data, and sends the final result to a base station directly. Two problems must be addressed to ensure that the base station obtains the correct result. First, the fusion node must correctly fusion all of the collected data. The second problem concerns assurance of the fusion result (transmission

between the fusion node and the base station is assumed herein to be error free). Since the fusion node may be compromised, forged data may be transmitted to the base station, which has no way to detect such forged results. This is the so-called stealthy attacks, where an attacker tries to trick the base station to accept a forged result [1].This work only focuses on the stealthy attack but not others. The main idea of the proposed scheme is to avoid using integrity mechanism such as Message Authentication Codes (MACs) which introduce extra transmission overhead. Since only stealthy attack is considered, for which assurance of receiving correct fusion data is the main issue, a simple voting scheme is proposed. The fusion result is broadcast such that compromised node can only jam the signal but cannot modify it. However, we do not consider the denial of service attack in this work. The only overhead that is taken into account is the extra effort for preventing the network from the stealthy attack.

## III. Proposed Architecture

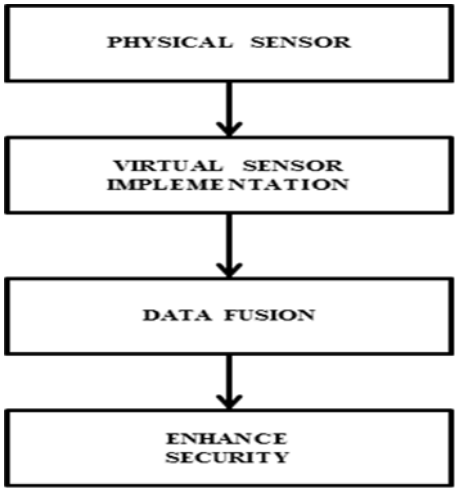


**Fig. 2 Proposed Architecture**

In this work, first the physical sensor is converted into virtual sensor. Later the data collected by sensor are fused based on data fusion technique. There is for possibility of intruders to attack or change the data sent by sensor. Security is increased in order to overcome threat and attack**.**

## IV. Virtual Sensor

Virtualization provides the ability to do several things. First, when associated with a platform, a virtual sensor or actuator can be created from physical devices. For example, if a platform has a thermistor that provides voltage readings as an output, a virtual sensor could be defined that would use the platform's processor to take thermistor output and, using calibration information, transform it to Celsius or Fahrenheit responses. Virtual devices can also be a collection of sensor outputs or actuator actions. There are two major types of virtual sensors/actuators: those focused on platforms and those focused on sensor fields. Platform virtual sensors/actuators are associated only with basic sensors and/or actuators on a physical platform. For example, sensor field can have a virtual sensor or actuator associated with it. A field virtual sensor is an aggregate virtual sensor that can take readings from all the same sensors in the field and use a function such as Average, Maximum, or Minimum as possible virtual sensor output. Virtual sensors can also be associated with groups of sensors in the sensor field. This creates subgroups of platforms that use a function to develop a composite value. For instance, consider a sensor field throughout a building. A field virtual sensor could be the temperature sensors in a room providing a single temperature reading for the room.

Consider a sensor network made up of platforms that have two sensors on them – represented in the diagrams as a triangle and a circle. Figure 3a. Shows a single platform with two sensors. Figure 3b shows a platform where a Virtual sensor is made by combining the output of two sensors. This would be a virtual sensor. The platform virtual sensor would have a function and a list of members in this case the types of sensors that make up the virtual sensor

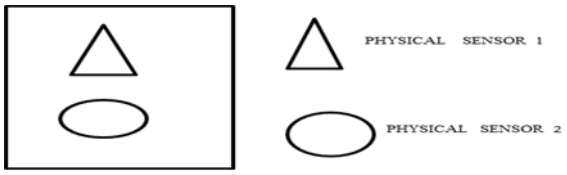


**Fig 3a. Two Physical Sensors**

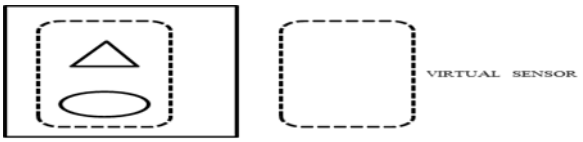


**Fig 3b. Two Physical Sensors made into Virtual sensor**

## V. Time Slotted Voting Scheme

The network structure for data fusion assurance in such VSNs with N sensors and $M_1$ fusion nodes is illustrated in fig 4. H hops are required to transmit the fusion result to the base station from the fusion node. At the $h_{th}$ hop, h = 1,2,. . . ,H - 1,$M_{h+1}$ fusion nodes1 are grouped at the h + 1 layer to receive and forward the fusion result. Note that the relation between $M_{h+1}$ and $M_h$ is arbitrary. Local time synchronization is assumed at each layer 2. The base station obtains the fusion result at the $H_{th}$ hop (the final hop). This network structure can be found in clustered WSNs [15]. Several real multi hop sensor networks can be found in [16]. In the direct-voting scheme, the base station consumes most power in the polling process. In this work, we propose a time-slotted voting approach, in which every fusion node at one layer of the multi-hop VSN transmits its vote or fusion result to the fusion node at the next layer in a pre-assigned and fixed time slot. With such pre-assigned time-slotted transmissions, no polling is necessary. We use the fusion process of the $h_{th}$ layer fusion nodes as an example for discussion. Fusion nodes of other layers follow the same procedure. Assume there

are $M_h$ fusion nodes at the $h_{th}$ layer. In general, each of the $M_h$ fusion nodes gets a chance to submit its data or vote, unless it is unnecessary. The transmission schedule for these $M_h$ fusion nodes can be rotated to balance their power consumption. All fusion nodes will listen while other fusion nodes at the same layer transmit. A threshold $T_h$ is used in order to decide whether a certain result has obtained enough votes. Without loss of generality, we name the $M_h$ fusion nodes at the $h_{th}$ layer as node 1, 2. . . i, . . . , $M_h$ according to the sequence of their transmission schedule. Therefore, node 1 sends first and node 2 sends next, and so on. When it is node i's turn to transmit, it will choose the first clause that agrees with its observation thus far:

C1. If more than $T_h$ votes have been submitted to support a certain fusion result, node i remains silent.

C2. If no fusion result has received at least $T_h$ - ($M_h$ - i) votes, node i remains silent.

C3. If there has been a fusion result transmitted earlier on (by one fusion node whose transmission schedule is ahead that of node i), node i will send an agreement vote to support this result.

C4. Otherwise, it will send its fusion result.

At the end of the transmission time slots of all fusion nodes at this layer, if there exists a data fusion result with at least $T_h$ supporting votes, this result will be accepted.

We explain the intuitive reasons to follow the different clauses in our algorithm as below. Our algorithm ensures that a data fusion node sends its data only when it is necessary to do so. The objective of each hop is to ensure an agreement on a correct data fusion result. If there has been such a result with more than $T_h$ supporting votes (including the original sender), other nodes do not need to vote (hence clause C1). In some scenarios, the nodes close to the end of the transmission sequence may see that, even if all the rest of fusion nodes agree with the currently most-popular result, there is no way to come up with a result with at least $T_h$ supporting votes.

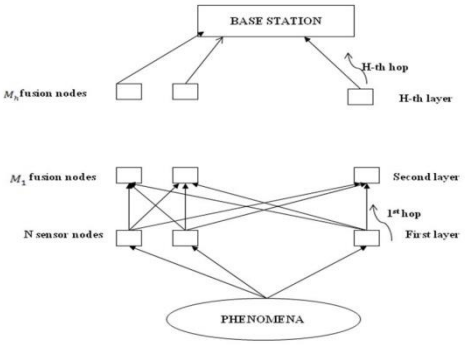


**Fig 4. Structure of multi hop sensor**

Therefore, it is useless to send in more votes or even new data (hence clause C2). Clause C3 simply states that a fusion node will send an agreement vote to support a result that has been submitted before. Clause C4 makes sure that someone will submit new results when there is a chance to obtain enough votes.

## VI. Enhancing Security

In our work, we first use the LCG to generate a random number $X_1$ (Step 1) and embed the pseudo-random number $X_1$ into the plaintext message (Step 2). We then apply the permutation function (Step 3). $X_1$ will also serve as the source of the permutation function. The final cipher text is obtained after Step 4.

a. Step 1 – Random Number Generation: We use the LCG to generate the random number. Given a 16 byte block cipher, one 16 byte random number, $X_1$, is needed.

b. Step 2 – Stage I: Suppose $P_1$ and $P_2$ are the plaintext message to be encrypted using this block cipher. Each pi is 8 bytes. We embed the pseudo-random number $X_1$ into the plaintext message in the following way. For example, let Wireless sensor (16 bytes) be the message to be encrypted. So $P_1$ = Wireless, and $P_2$ = sensor. The first three characters of $P_1$ are W = 87,i = 105, and r =114. The embedding operations are simply the addition modulo 256. If

$X_1$ = 10 5A FB 11 FC BB 00 11 22 33 44 55 66 77 88 $99_h$

The values of the first three bytes are $10_h$ = 16, $5A_h$ = 90, and $FB_h$= 251. Therefore, the values of the first three ciphertext characters encrypted are:

87 + 16 mod 256 = 103
105 + 90 mod 256 =195
114 + 251 mod 256 = 109

$C_1$, and $C_2$ are the scrambled text after $X_1$ is embedded. Each $C_i$ is also 8 bytes.

a. Step 3 – Permutation: $X_1$ is broken into 16 1 byte random numbers, denoted as $B_0, B_1, \ldots\ldots\ldots B_{15}$, respectively. We introduce a permutation function P over $Z_{16}$={0,1,2,…….15}.

b. Let values $\pi = \pi_0\pi_1\pi_2\ldots\ldots\ \pi_{15}$ be constructed as follows

I. $\pi_0 = B_0$ mod 16;

II. $\pi_i$ =(n mod 16), for i=1 . . . 15 with n is the smallest integer such that n ≥ $B_i$ and $\pi_i$ /∈ {$\pi_0, \pi_1\ldots\ldots\ldots\pi_{i-1}$}

c. Step 4 – Stage II: After we obtain Π, we apply Π to C1C2 obtained in Step 2 in a standard manner, i.e., the $i^{th}$ byte of Π(C1C2) is the Π $i^{th}$ byte of C1C2. Presented by 8 byte segments, let Π(C1C2) =Cl1Cl2, which are our final encrypted message.

Decryption is straightforward. The receiver node could generate the same $X_1$ that the sender generates. Using $X_1$, the receiver can obtain $P_1$ and $P_2$ .Based on an LCG based block

cipher, sensor nodes, such as nodes A, B, C, and D have monitored some events and transferred the readings to their immediate aggregator, node H. Each sensor node appends a MAC to the plaintext message P and uses their shared secret keys with H to encrypt the whole message. After H receives the readings, H uses the corresponding secret to decrypt and to authenticate the received messages. This time, node H appends a new MAC to the aggregated result and uses its shared secrets with its immediate aggregator, node J, to encrypt the whole message. The process continues until the result reaches the base station

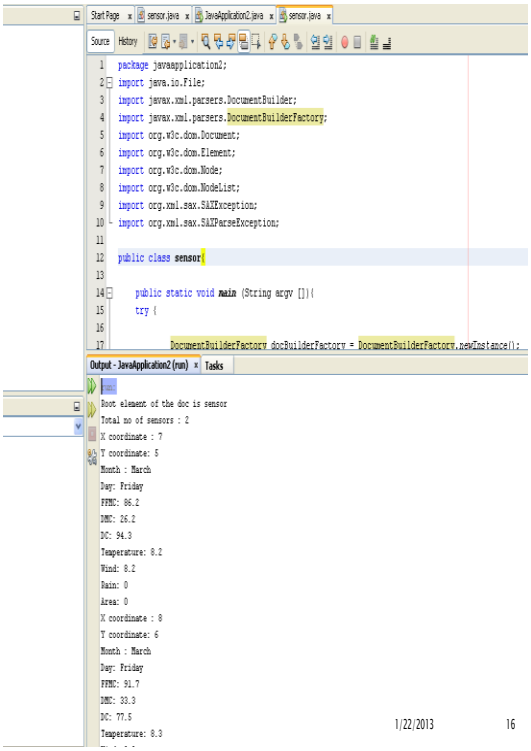


**Fig 5. Result**

The result for virtual sensor is shown in fig 5. The physical sensor is made into virtual sensor and data fusion process is performed. Overall security of data sending from sensor is increased.

## VII. Conclusion and Future Work

This paper presented data fusion and security techniques in virtual sensor network. We investigate the information assurance issue of the data fusion process, in which the compromised

Sensor nodes may launch stealthy attacks to trick data fusion nodes and eventually the base station to accept false results. The proposed scheme provides good security against sensor node compromise. Moreover, the traffic to be transmitted at $h_{th}$ hop is $O(T_h)$. In Future work It is worthwhile to find other applications such as encryption on some virtual sensor network, where the communications between sensors are short and frequent, and the computational resource on each sensor is limited.

## References

[1] Hung-Ta Pai, Jing Deng, Yunghsiang S. Han, "Time-slotted voting mechanism for fusion data assurance in wireless sensor networks under stealthy attacks", Computer Communications 33 (2010) 1524–1530,2010

[2] Y. Lin, B. Chen, P.K. Varshney, "Decision fusion rules in multi-hop wireless sensor networks", IEEE Transactions on Aerospace and Electronic Systems 41 (2) (2005) 475–488,2005

[3] Nathan Ota, William T.C. Kramer, "TinyML:Meta-data for wireless sensor networks". http://dnclab. keley.edu/~ nota/research/TinyML/TinyML2.htm, 2005

[4] Sanem Kabaday, Christine Julien, William O'Brien, and Drew Stovall, "Virtual Sensors: A Demonstration", TR-UTEDGE-2007-003,2007

[5] Kabadayi S., Pridgen A., Julien C., "Virtual Sensors: Abstracting Data from Physical Sensors". Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks; Buffalo-Niagara Falls, NY ,USA 26–29 June ,2006.

[6] Bo Sun, Yang Xiao, Chung Chih Li, Hsiao-Hwa Chen d, T. Andrew Yang, "Security co-existence of wireless sensor networks and RFID for pervasive computing", Computer Communications 31 (2008) 4294–4303,2008

[7] TIAN Bin, YANG Yi-xian, LI Dong, LI Qi, XIN Yang, "A security framework for wireless sensor networks", December 2010, 17(Suppl. 2): 118–122,2010

[8] Md. Motaharul Islam, Mohammad Mehedi Hassan, Ga-Won Lee, and Eui-Nam Huh, "A Survey on Virtualization of Wireless Sensor Networks", Sensors (Basel). 2012; 12(2): 2175–2207,2012

[9] Islam M.M., Hasan M.M., Huh E.N, "Virtualization in Wireless Sensor Network: Challenges and Opportunities". Proceedings of the 13th International Conference on Computer and Information Technology (ICCIT); Dhaka,Bangladesh. 23–25 December,2010.

[10] W.Stallings "Wireless Communications and Networks", Prentice Hall, Upper Saddle River, NJ, 2002

[11] H. Chan, A. Perrig, B. Przydatek, D. Song ,SIA: "secure information aggregation in sensor networks", Journal of Computer Security 15 (1)(2007) 69–102,2007

[12] H. Chan, A. Perrig, D. Song,"Secure hierarchical in-network aggregation in sensor networks", in: Proc. of ACM CCS '06, Alexandria, VA, Nov. 2006, pp.278–287

[13] K.B. Frikken J.A. Dougherty IV, An efficient integrity-preserving scheme for hierarchical sensor aggregation, in: roc. of ACM WiSec '08, Alexandria, VA, 2008, pp. 68–76.

[14] O.Younis, M. Krunz, S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges", IEEE Network 20(3) (2006) 20–25.

[15] K. Römer, F. Mattern, "The design space of wireless sensor networks", IEEE Transactions on Wireless Communications 11 (6) (2004) 54–6

[16] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TinyDB: "An cquisitional query processing system for sensor networks. ACMTrans. on Database Systems", 30(1):122–173, 2005

[17] Y. Yao "The cougar approach to in-network query processing in sensor networks", ACM SIGMOD Record, 31(3):9–18,2002.

## Authors

**First Author** – The author Mr.A.Vigneshkumar1 is currently working as an Assistant Professor in the Department of Computer Science and Engineering in Bannari Amman Institute of Technology. He has completed his UG & PG in Anna University, Chennai. He has published 1 paper in International Conference and 2 papers in National conference. His area of research interests are Computer Networks, Cloud computing, big data, XML and Web Services

and Data Mining. He has attended 3 workshops and seminars., Email: vigneshkumararun@gmail.com

**Second Author** – The author Mr.V.Balamurugan2 is currently working as an Assistant Professor in the Department of Computer Science and Engineering in Bannari Amman Institute of Technology. He has completed his UG & PG in Anna University, Chennai. He has published 6 Research papers in Highly Impact factor Journals. He has published 2 papers in International Conference and 4 papers in National conference. He is a member of various professional bodies like ISTE, CSI, IAENG, UACEE and SAI. His area of research interests are Web Technology, XML and Web Services, Data Mining and Data Structures. He is one of the university rank holder during the Academic year 2012-2014. He has attended 4 workshops and seminars., Email: balamuruganvsrit@gmail.com