Autoregressive integrated moving average model–based secure data aggregation for wireless sensor networks

Hongtao Song, Shanshan Sui, Qilong Han, Hui Zhang and Zaiqiang Yang

Abstract
Nodes in a wireless sensor network are normally constrained by hardware and environmental conditions and face challenges of reduced computing capabilities and system security vulnerabilities. This fact calls for special requirements for network protocol design, security assessment models, and energy-efficient algorithms. Data aggregation is an effective energy conservation technique, which removes redundant information from the data aggregated from neighbor sensor nodes. How to further improve the effectiveness of data aggregation plays an important role in improving data collection accuracy and reducing the overall network energy consumption. Unfortunately, sensor nodes are normally deployed in an open environment and thus are subject to various attacks conducted by adversaries. Consequently, data aggregation brings new challenges to wireless sensor network security. In this article, we propose a novel secure data aggregation solution based on autoregressive integrated moving average model, a time series analysis technique, to prevent private data from being learned by adversaries. We leverage the autoregressive integrated moving average model to predict the data volume in sensor nodes, and update and synchronize the model as needed. The experimental results demonstrate that our model provides accurate predictions and that, compared with competing methods, our solution achieves better security, lower computation and communication costs, and better flexibility.

Keywords
Wireless sensor networks, secure, data aggregation, autoregressive integrated moving average, Internet of Things

Date received: 7 January 2020; accepted: 24 February 2020

Handling Editor: Xiaojiang Du

Introduction
As a distributed sensor network, wireless sensor networks (WSNs) have been widely used in many different domains, such as environmental monitoring, intelligent transportation, medical care, and military affairs. As shown in Figure 1, a WSN combines the information world with the real world to realize the convenient Internet of Things (IoT) project. In the harsh wild, people cannot stay for a long time, and there are no fixed network facilities for communication. In contrast, due to their self-adaptive and self-organizing properties, ad hoc WSNs are particularly suitable for applications in such situations. A massive number of sensor nodes can be deployed to collect and process monitoring data and send it to a remote control terminal by wireless...
communication. Since WSNs are normally deployed in unattended open environments with limited resources and are of a large network scale, there are many security challenges in data sensing, collection, transmission, and processing. Among them, data aggregation and network security are two important hot research topics. Due to a large number of nodes and a large amount of data generated in WSN, end-to-end security is difficult to achieve. Nodes in WSN have a limited communication range and noise or collision may cause transmission failure or error. Considering the aggregated data, privacy protection is necessary. Attackers will attempt to obtain or tamper with private data using link eavesdropping and nodes compromising. For private data protection, methods such as data perturbation, anonymity, or encryption can generally be used to accomplish data aggregation or query with no leakage of private data.

Data aggregation can internally aggregate raw data collected from sensor nodes and eliminate redundant data. It reduces the number of transmissions and communication costs and thus extends a network’s overall lifecycle. In the literature, there are two ways of realizing data aggregation in WSNs. The first way is to improve raw data integrity and security. WSN nodes are often affected by some environmental factors, and the data collected by sensor nodes may be imprecise, noisy, and sometimes even fake. This requires data authentication to ensure integrity and security. Although the authentication method improves data validity, it leads to a longer aggregation time and increased energy consumption and reduces a network’s aggregation capability. Some communication protocols have better data aggregation capability, but lack considerations on system security. The second is to utilize secure aggregation algorithms. Due to the similarity of the data sampled from neighbor nodes, the impact of some individual malicious data can be reduced by an aggregation processing on raw data. This line of research also has some limitations. The aggregation nodes cannot always obtain multiple non-redundant valuable data points, and the effect may be different in various industrial applications. Data aggregation can obtain better results in time-driven applications, for example, air pollution index monitoring. In contrast, it is less effective for event-driven applications, such as fire alarm monitoring and positioning, and might even mishandle some important information.

In terms of network security, asymmetric encryption is not suitable for WSNs, and symmetric encryption cannot solve all security problems as well. There is a plethora of factors that can affect the security of WSNs. When designing a network system model, how to evaluate its overall security and how to avoid data damage or loss and system crashes are of practical importance.

WSN nodes are usually composed of low-power sensors equipped with antennas and small computing units, which are used to monitor the surrounding environment and transfer data to the base station. For a battery-powered sensor node, the biggest challenge is how to reduce energy consumption as much as possible and extend the lifecycle of the network because computation and communication consume the most energy.
Rault et al.\textsuperscript{1} listed energy-saving measures in WSNs from different perspectives, including energy collection and transmission. Among all the approaches, data aggregation was considered to be an effective energy-saving method. Data aggregation integrates the data of sensor nodes and returns the required or valuable data to users. The key of data aggregation is to remove redundant data transmitted in WSNs, which reduces the energy cost by cutting down the amount of data transmitted.

Networks might suffer from internal and external attacks in the data aggregation process.\textsuperscript{2} Researchers have proposed several effective encryption schemes for WSNs, which can effectively resist external attacks, but need to consider the problem of energy consumption. Since attackers will try to compromise sensor nodes, internal attacks are generally more harmful. They may lead to the leakage of private data and keys. Furthermore, attackers can tamper with sensor nodes’ private data, leading users to wrong aggregation nodes, and in turn directly mislead users’ decision making in the corresponding scenario. Traditional data aggregation algorithms include Bayesian probability theory,\textsuperscript{3} Dempster–Shafer algorithm,\textsuperscript{4} Markov random field theory, weighted average method, and Kalman filter algorithm. Some of the recent research results focus on improving the limitations of these traditional algorithms. With the development of relevant technologies, some researchers introduced new theories, such as information entropy,\textsuperscript{5} fuzzy theory,\textsuperscript{6} and neural network,\textsuperscript{7} to data aggregation in order to improve aggregation accuracy.

Madden et al.\textsuperscript{8} presented the tree structure–based tiny aggregation (TAG) protocol for aggregation in low-power distributed wireless environments. In this protocol, a WSN is considered as a connected graph whose nodes are the sensor nodes. At the beginning of data aggregation, each child node transfers data to the base station along the layers in the aggregation tree. Avokh and Mirjalily\textsuperscript{9} proposed a dynamic balanced spanning tree (DBST) for data aggregation, which considers several criteria to balance the energy consumption among sensor nodes. DBST not only minimizes the maximum energy consumption among the sensor nodes, but also balances the traffic load in the network by a dynamic design of the routing tree. Based on locality sensitive hashing (LSH) algorithm, Patil and Kulkarni\textsuperscript{10} introduced support vector machine (SVM) to reduce data redundancy and eliminate the abnormal sensor nodes sending wrong data. Mantri et al.\textsuperscript{11} presented a solution for effective data gathering with inner-network aggregation using the bandwidth efficient heterogeneous aware cluster based data aggregation (BHICDA) algorithm. It considers a network with heterogeneous nodes in terms of energy and mobile sink to aggregate data packets. A novel cluster-based strategy for eliminating redundancy in the data dissemination process was proposed by Ramachandran et al.\textsuperscript{12} Obviously, for multi-node prediction, the overall computation cost in the network significantly increases, but the amount and frequency of communication are largely reduced.

Tulone and Madden\textsuperscript{13} proposed a method to approximate the values of sensors in a WSN based on autoregressive (AR) models. When the prediction error exceeds the threshold value of 13, a node synchronizes data with the base station and transmits new data. This scheme does not take into account the time slot of data collection and transmission. Frequently updating the model will lead to degraded network service quality and high delay. Moreover, updating the model of a node requires interactions with the base station. When the node is far away from the base station, it will cause high communication overhead.

Lu et al.\textsuperscript{14} proposed a distributed data aggregation scheme based on an autoregressive moving average (ARMA) model. It sets an observation window and measures the prediction error according to the size of the observation window. Similarly, autoregressive integrated moving average (ARIMA) is also used to model historical data.\textsuperscript{15,16} For data produced in WSNs, ARMA proposed for stationary time series can reach high accuracy, but the complexity of ARIMA brings additional computation overhead to the sensor nodes. Based on the AR model, Ghaddar et al.\textsuperscript{17} adjusted the coefficients of the model using predicted residuals to avoid the process of multiple modeling processes. Unfortunately, all the above prediction-based data aggregation schemes do not consider data privacy, and thus are vulnerable to privacy attacks.

In the context of data aggregation, security requirements are mainly reflected in terms of data confidentiality, integrity, and immediacy. Most of the early secure data aggregation schemes made use of hop-by-hop encryption,\textsuperscript{18} where each packet must experience a decryption–aggregation–encryption process at the next hop. If an intermediate node is compromised, the private data in itself and its children may leak. Moreover, the nodes execute encryption and decryption operations when sending data packets to the base station, which brings computation overhead and network delay. The end-to-end secure data aggregation scheme well solves the above problems.\textsuperscript{19} In this scheme, once data are encrypted, any intermediate node cannot decrypt it. Only after the base station receives an aggregated data packet, it can decrypt the data to extract the plaintext information. However, it cannot prevent attackers from actively tampering with data or injecting fake packets into the network.

Privacy homomorphism was proposed by Rivest et al.,\textsuperscript{20} which directly performs calculations on encrypted data. It well fits WSNs’ end-to-end
confidentiality protection requirements. Castelluccia et al.\textsuperscript{21} proposed a simple and provably secure additively homomorphic stream cipher (CMT) that allows efficient aggregation of encrypted data. The key is produced using a pseudorandom number generator, making the scheme more feasible for WSNs. Based on CMT, Papadopoulos et al.\textsuperscript{22} devised the secure in-network processing of exact SUM (SIES) queries method. It leverages the idea of secret sharing to protect the integrity of a data aggregation process and adds the aggregation cycle count to protect data instantiation. Li and Gong\textsuperscript{23} proposed a network coding scheme based on the homomorphic message authentication code (MAC) for integrity protection. It requires each leaf node to share a key with the base station. However, if a leaf node is compromised, its key information will be unveiled, which further jeopardizes the entire network’s security. Homomorphic encryption-based data aggregation schemes have drawbacks such as malleability, unauthorized aggregation, and limited aggregation functions. To solve these problems, Zhong et al.\textsuperscript{24} proposed a secure data aggregation scheme by combining homomorphic encryption technology with a signature scheme. Gopikrishnan and Priakanth\textsuperscript{25} proposed a hybrid secure data aggregation (HSDA) to provide high secure data aggregation in an energy-efficient way, which implements an end-to-end symmetric key cryptography for secure authentication using shared public key and uses hop-by-hop asymmetric key cryptography with the private keys of each node for data integrity and confidentiality.

Being the continuation of prediction-based secure data aggregation, the exponential smoothing data aggregation (ESDA)\textsuperscript{26} introduced the exponential smoothing method to time series processing, which has the advantages of less transmission and higher security. He et al.\textsuperscript{27} presented a cluster-based private data aggregation (CPDA) scheme by introducing the secure multi-party computation (SMC) to data aggregation in WSNs. First, the sensor nodes are formed randomly into clusters. Then, after encrypting and cross-summing, the data collected by the sensor nodes are transmitted to the corresponding cluster heads. Finally, data aggregation in cluster heads and among the clusters will be executed, and the final results are sent to the base station. The CPDA scheme can effectively preserve data privacy, but lead to the increasing computation and communication overhead. In order to reduce the communication overhead, Zhang et al.\textsuperscript{28} proposed a privacy-preserving data aggregation protocol called rotation-based privacy-preserving data aggregation (RPDA), which is suitable for additive aggregation. The protocol protects the actual data from other nodes based on a rotation scheme and achieves accurate aggregation results. Considering IoT-enabled applications, Yu et al.\textsuperscript{29} proposed a cluster-based data analysis framework using recursive principal component analysis, which can aggregate the redundant data and detect the outliers in the meantime. Vinodha and Anita\textsuperscript{30} presented a comprehensive survey on existing privacy preserved data aggregation techniques for WSN, which explored the various mechanisms for data aggregation for preserving energy of sensor nodes by eliminating the redundant data transmission. It was indicated that the topology of nodes had a significant impact on the performance of the data aggregation schemes. By comparing and analyzing on artificial intelligence (AI)-based data aggregation techniques in WSNs, Kumar et al.\textsuperscript{31} designed a modified protocol, which is better in terms of network lifetime and throughput of the networks.

To sum up, data aggregation is an important means to reduce the amount of data transmission, which can indirectly improve network lifecycle and bandwidth utilization. In the process of data aggregation, privacy protection is the core, which requires us to reduce data traffic, computation cost, and energy consumption while preserving data privacy.

In this article, we introduce the ARIMA model, a technique for time series analysis, to data aggregation, and propose an improved data aggregation scheme, which offers high security, low computational and communication costs, high accuracy, and better flexibility. The experimental results confirm the desirable performance of our ARIMA model–based solution.

The rest of the article is organized as follows. In section “System model,” we introduce the network model, security model, and attack model. In section “Proposed method,” we describe the aggregation method and privacy protection strategy based on the ARIMA model. The experimental results are presented in section “Experimental results and analysis.” Finally, we conclude the article in section “Summary and outlook.”

\section*{System model}

\subsection*{Network model}

Sensor nodes in a network can be divided into three categories, namely the base station, intermediate nodes, and leaf nodes. The base station is located on the top of a WSN, which is responsible for connecting terminal devices and sending aggregated (query) results to users. An intermediate node, also called a cluster head, is in charge of processing data within the cluster. A leaf node collects and sends raw data.

Different from a tree structure, the intracluster relationship in this scheme is a planar data link, and we mainly consider the sum calculation in this article. With slight modifications, the model is also suitable for more complicated aggregation operations, for example, computing the mean and the variance.
A query operation is defined as \( Q = (\text{class} = f) \cap (\text{attribute} = a) \), where \( f \) is the current aggregation operation and \( a \) denotes the specific parameter in this query.

The network model has the following characteristics:

1. The \( N \) sensor nodes are randomly distributed in a region.
2. Each node has the same communication range \( R \) and sensing ability.
3. Each node has the same initial energy sufficient to support the proposed scheme.
4. The communication energy cost of each node is different.
5. The base station knows the exact location of each node.

The proposed model is based on a topology of TAG tree aggregation, and the aggregation process can be divided into three steps. First, non-cluster head nodes use the ARMA model to model historical data and transmit the model parameters and historical data to the cluster head. Then, the cluster head calculates the data of each node at the next time and waits for messages from these nodes. Finally, the cluster head performs data aggregation processing as needed.

**Security model**

This scheme is based on the semi-honest model, that is, the participating nodes follow the protocol completely, but they will attempt to obtain the private data of other participating nodes. Meanwhile, an attacker will try to eavesdrop the original data and compromise some nodes to tamper with the aggregated data.

In this article, we use the key distribution mechanism, which includes key pre-distribution, shared key discovery, and key pair establishment. In the key pre-distribution stage, a key pool with \( K \) keys will be generated. For each node, they randomly select \( K \) keys from the key pool and form a key ring. During the shared-key discovery step, each node finds the neighbors with the same key through information exchange. If two neighbor nodes hold a common key, a secure connection will be established between them. In the key pair establishment step, if neighbor nodes do not have a common key but are interconnected by multi-hop secure paths, the path keys are formed between them.

Data aggregation should satisfy three requirements: privacy, accuracy, and efficiency.

**Attack model**

An attacker may monitor the wireless channel to obtain a node’s private data or compromise an internal sensor node to tamper with uploaded data. To this end, the purpose of our proposed method is to protect data privacy and determine whether an internal node has been tampered with according to a threshold. Moreover, the energy cost also needs to be considered.

**Proposed method**

Data collection is to transmit sensory data from multiple sensor nodes to the base station for further processing. Due to limited energy, sensor nodes usually do not send data directly to the base station, and the sensing data of each sensor node have a certain extent of redundancy and correlation.

In Figure 2(a), if all the nodes upload collected data directly to their parent nodes without any processing, the closer a node is to the base station, the more packets it needs to transmit. In contrast, if an intermediate node is allowed to aggregate the data sent from its child nodes, as illustrated in Figure 2(b), the number of packets required to be sent by upper nodes can be greatly reduced.

Data aggregation is to eliminate redundant data packets and utilize data correlation to reduce the total amount of data transmission in a WSN, and then reduce the probability of data collision and congestion in the network, and finally achieve the goal of saving resource consumption and extending the lifecycle of the entire network.

**ARIMA model**

The functional expression of the model is described by \( ARIMA(p, d, q) \), which can be considered a combination of the AR and moving average (MA) model. The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lags. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.
The I (for “integrated”) indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once). The purpose of each of these features is to make the model fit the data as well as possible. The symbols $p$, $d$, $q$, which are non-negative integers, refer to the AR (i.e. the number of time lags of AR model), differencing (i.e. the number of times the data have had past values subtracted), and MA (i.e. the order of MA model) terms of the ARIMA process, respectively. In the process of secure aggregation, the ARIMA model considers not only the temporal dependence of network nodes’ states, but also the interference of random fluctuations. Thus, it has a high prediction accuracy for short-term network trends.

As an important method to study time series, the ARIMA model is a typical method to model the rational spectrum of a stationary stochastic process, which can be used to solve many practical problems. Compared with the AR and MA model, ARIMA has more accurate spectral estimation and better spectral resolution performance, but its parameter estimation is relatively complex. Therefore, in engineering practice, AR and MA parameters are usually estimated separately to obtain a suboptimal scheme. Although AR and MA parameters cannot be estimated at the same time to identify the optimal parameter setting, the computation cost can be considerably reduced.

The most important assumption of the ARIMA model is the stationarity of time series data. The stationarity requires that the fitting curve obtained from the sample time series can continue along with the existing form inertia in the near future, that is, the mean and variance of the data should not change too much in theory. The stationarity can be divided into two categories: strict stationary and weak stationary. Strict stationary means that the distribution of data does not change over time; weak stationary means that the mean value and correlation coefficient of data will not change. In the process of data aggregation, strict stationary is too idealized and theorized, and weak stationary applies to most cases. For unstable data, we need to first stabilize it. The difference method is the most commonly used technology, which calculates the difference between time $t$ and time $t-1$ in a time series for generating a more stationary time series.

**AR model**

The AR model specifies that the output variable linearly depends on its own previous values and on a stochastic term. When analyzing the states of network nodes, it describes the relationship between current and historical values and leverages a variable’s historical events to predict the variable’s future events. The model is in the form of a stochastic difference equation, which is given as follows

$$y_t = c + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t$$

(1)

Here, $y_t$ is the current value, $c$ is a constant, $p$ is the order, $\gamma_i$ is the autocorrelation coefficient, and $\varepsilon_t$ is the error value (usually white noise).

There are four limitations when using AR models:

1. The model uses its own data for prediction, that is, the data used for training and predictions are the same.
2. The data used must be stationary.
3. The data used must be partial autocorrelative. If the partial autocorrelation coefficient is less than 0.5, the AR model is not suitable.
4. The AR model can only be used to predict the phenomenon related to its earlier states.

**MA model**

In time series analysis, the MA model is a common approach for modeling univariate time series, which specifies that the output variable depends linearly on the current and various past values of a stochastic (imperfectly predictable) term. Contrary to the AR model, the finite MA model is always stationary. The MA model focuses on the accumulation of error terms in the AR model and can effectively eliminate the random fluctuations in the prediction. It can be formulated as follows

$$y_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

(2)

Here, $\mu$ is the mean of the series, $\theta_i$ is the temporal average correlation coefficient, $q$ is the order, and the rest symbols are the same as those in the AR model.

**ARMA model**

In the statistical analysis of time series, ARMA models provide a parsimonious description of a (weakly) stationary stochastic process in terms of two polynomials, one for the AR and the other for the MA. The ARMA model can be given by

$$y_t = c + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

(3)

Here, $p$ and $q$ are the AR order and MA order, respectively, and they need to be defined manually. $\gamma_i$ and $\theta_i$ are the correlation coefficient in the AR and MA models, respectively, which need to be calculated. The rest
symbols are the same as those in the AR model. If the original data do not meet the stationarity requirement and are processed by the difference method, we can apply the ARMA model to the differenced data, leading to the ARIMA model.

**Differencing**

Differencing in statistics is a transformation applied to time series data in order to make it stationary. The difference between consecutive observations is computed as follows

\[ \Delta x_t = x_t - x_{t-1} \]  \hspace{1cm} (4)

where \( \Delta \) is the first-order difference operator. Similarly, the second-order differencing can be calculated by

\[ \Delta^2 x_t = \Delta x_t - \Delta x_{t-1} \] \hspace{1cm} (5)

In general, we can define the \( d \)th order difference in a similar way. If a random process contains \( d \) unit roots, it can be transformed into a stationary ARMA process after \( d \) difference. This random process is called the ARIMA process.

After the \( d \)th order difference, we need to determine whether the random state of the network is stationary at this time. For a suitable \( d \), \( x_t \) will be converted to the stationary stochastic process \( \Delta^2 x_t \), which is built as the ARMA process, denoted by ARMA(\( p, q \)). Then, the process representing random state \( x_t \) as ARIMA is completed. This ARIMA process is recorded as ARIMA(\( p, d, q \)). Actually, the construction of an ARIMA model mainly includes finding the \( d \)th order difference and estimating ARMA model parameters. The number of times of difference should not be too large; otherwise, the fluctuation will be too large. The parameters of an ARMA model are estimated by maximum likelihood.

In addition, the autocorrelation between the states of network nodes is to compare the ordered random state sequence with itself, and it reflects the correlation between the values of the same sequence in different timestamps. The partial autocorrelation function (PACF) calculates the partial correlation between two variables, which eliminates the interference of other intermediate variables.

For a stationary AR(\( p \)) model, when calculating the autocorrelation coefficient \( P(k) \) with a lag of \( k \), the actual result is not the correlation between \( x_t \) and \( x_{t-k} \), because there are \( k-1 \) variables between the two variables, which may produce an effect on the autocorrelation coefficient, and the \( k-1 \) variable itself is related to \( x_{t-k} \). This is a significant interference to the calculation of autocorrelation coefficient \( P(k) \), but the PACF can eliminate these interferences.

**Aggregation process based on ARIMA model**

Sensor nodes continuously monitor data continuously and update the ARIMA model when the prediction error is greater than the predetermined threshold. It has a few advantages. First, nodes can detect outliers exceeding the predetermined error range, which may be caused by nodes’ wrong observations or the change of the distribution of observations. Second, it does not need communication during the modeling step, and in the prediction stage, communication between a sensor node and the cluster head is needed only when the prediction error is too large. Third, the cluster heads utilize the model to calculate the predicted value at the next timestamp iteratively after receiving the data and model parameters from each sensor node. The model is updated only when the error exceeds the predetermined threshold, and the updated model is used for future predictions. The detailed aggregation process is illustrated in Figure 3.

To select a proper error threshold, we define the parameters in Table 1. According to the threshold for the upper limit of error, when \( th_{up} = \gamma \sigma(w) \), the maximum value of the probability that the observed value does not fall into the interval \([pre_t - th_{up}, pre_t + th_{up}]\) at time \( t \) is \( 1/\gamma^2 \), where \( \gamma \) is a real number greater than 1 and \( \sigma(w) \) is the standard deviation of white noise process.

At any time \( t \), the prediction error obeys the normal distribution with the mean value of zero and the standard deviation of \( \sigma(w) \). Following Chebyshev’s inequality, we have

\[ P(|pre_t - th_{up}| \geq th_{up}) \leq \frac{\sigma(w)^2}{th_{up}^2} = \frac{\sigma(w)^2}{\gamma^2\sigma(w)^2} = \frac{1}{\gamma^2} \] \hspace{1cm} (6)

Here, we can choose an appropriate \( \gamma \) (e.g. 4 or 5) to make the error probability \( 1/\gamma^2 \) negligible or acceptable. \( th_{down} \) needs to be set according to specific situations, but it must be greater than 0 and less than \( th_{up} \). If we set \( th_{down} \) to a small value, the acceptable error of the model will also be smaller. In this case, sensor nodes need to update the model frequently and communicate with the cluster head, which results in additional computation and communication overhead. If we set \( th_{down} \) to a large value, the number of model updates will be small. However, cluster heads will not store the observation data of each node at each timestamp. Over time, the cluster heads basically use prediction data to make further prediction, making the data trend linear. In this case, predictions are not meaningful any more.

We divide error calculations into two categories based on the time interval between two data observations. A cluster head needs to wait for a period of time to receive new data and model parameters to update the model. If the time gap \( \tau \) is small, data need to be
collected frequently in practice, and the waiting time in the network is long. As a result, the freshness of data cannot be guaranteed. Therefore, when the time gap is small, after collecting $W$ samples and building a model, we only calculate the error in every $S$ time windows. The error could be measured in terms of mean squared error (MSE) and mean absolute error (MAE). When the time gap is large, we compute the error for each prediction. The pseudo code for building and updating the model is given in Algorithm 1.

Considering the limited storage in sensor nodes, each node only stores the latest $W$ data, where $W > S$. If the time gap of data acquisition in the network is small, each sensor needs to acquire additional memory space to store the latest $S$ prediction values for error calculation. The proposed method also considers the privacy of data aggregation in that it prevents the data transmitted in the wireless channel from being eavesdropped. Specifically, starting from the first model update, a node needs to anonymize the data sent to the cluster head. Take an ARIMA(2, 1, 2) model as an example. A

![Diagram](image_url)

**Figure 3.** The aggregation process based on the ARIMA model.

**Table 1.** ARIMA model parameters.

| Parameter | Description                                           |
|-----------|-------------------------------------------------------|
| $\tau$    | Time interval between two data observations           |
| $th_{\text{down}}$ | Threshold needed to trigger a model update              |
| $th_{\text{up}}$ | Threshold for the upper limit of error                  |
| $W$       | Number of samples required for modeling                |
| $S$       | Size of an observation window                          |
| $pre_t$   | Predicted value at time $t$                            |
| $sent_t$  | Observed value at time $t$                             |

ARIMA: autoregressive integrated moving average.
node needs to send the latest two observations to the cluster head, which are considered the node’s private data. In our solution, the node only sends the residuals, which are the absolute values of $pre^i$ and $sent^i$, along with a private random number to the cluster head. The cluster head can obtain the real observed value by removing the private random number and adding the predicted value.

### Experimental results and analysis

In the experiments, an Ubuntu server with the Hadoop platform is used to simulate a distributed WSN. The simulation analysis of the proposed framework is conducted on the PAMAP2 Physical Activity Monitoring dataset\(^{23}\) in the UCI machine learning repository. The dataset contains data of 18 different physical activities, performed by nine subjects wearing three inertial measurement units and a heart rate monitor. It is multivariate and collected from real sensor nodes. There are 3,850,505 instances in the dataset, each of which contains 52 attributes. To evaluate the performance of the method proposed in this article, we compare it with ESDA, TAG, CPDA, and RPDA. The RPDA was the first to propose a chain aggregation scheme in clusters.

**Privacy**

In the experiments, we propose a key agreement scheme for the ARIMA model. First, $k$ arbitrary encryption keys are selected from the key pool with a total of $K$ keys. Any two nodes have at least one chance to share a common key, so the connectivity can be described by

$$P_{connect} = 1 - \frac{[(K - k)!]^2}{(K - 2k)! \cdot K!}$$  

Then we set a random node to receive the corresponding encrypted message, whose probability is $P_{overhear}$. It also means that there is the third node with the common key, and the probability of its existence is

$$P_{overhear} = \frac{k}{K}$$  \hspace{0.5cm} (8)

We assume that there are 10,000 keys in the key pool. In the stage of key distribution, 200 nodes are selected from the key pool. Hence, $P_{connect} = 98.3\%$ and $P_{overhear} = 0.2\%$, which shows that the probability of detecting encrypted data is negligible. Therefore, our method can resist common attacks including intercept and collusion. Moreover, the ARIMA model can also effectively defend trapping and tampering.

ARIMA controls the maximum allowable error within a small range by adjusting the parameter $\gamma$ (see equation (6)). When aggregating data, the cluster head can consider that a node has been compromised and the attacker is tampering with the data if the cluster head detects that the node frequently updates the parameters, and that the residuals sent are extremely abnormal (e.g. they exceed three times the standard deviation of the mean). Hence, the WSN will refuse to receive data from the node and will not predict the state of the node.

**Computation cost**

Since ARIMA is a strongly dynamic system, we assume that a node performed $N$ aggregations and its model is updated $N_u$ times. To build a model for a non-cluster head node, we need to solve a system of linear equations. When the model does not need to be updated, a polynomial calculation consisting of four multiplication operations and five addition operations is performed. When the model needs to be updated, we need to solve an additional system of linear equations and perform an addition operation to calculate the residuals sent to the cluster head.

The cluster head makes a prediction at each timestamp. It involves a polynomial operation and an extra addition operation to update the model. The computational cost $C_{mem}$ for a non-cluster head is

$$C_{mem} = N_u \times (C_{eq} + C_{add}) + (N - N_u) \times (4C_{mul} + 5C_{add})$$  \hspace{0.5cm} (9)

where $C_{eq}$, $C_{add}$ and $C_{mul}$ are the computation cost for solving the system of linear equations, average byte addition operations, and average byte multiplication operations, respectively. Then, the computation cost $C_h$ for a cluster head is

$$C_h = (n - 1) \times (N \times C_{add}) + (N - N_u) \times (4C_{mul} + 5C_{add})$$  \hspace{0.5cm} (10)
where \( n \) denotes the number of the pseudo data, which is generated by a quadratic function of the observed data of each node, a private random number, and a public seed number.

It can be seen that the computation cost of both cluster head nodes and non-cluster head nodes completely depends on the order of the ARMA model and the number of model updates. The higher the order of the model, the higher the precision and the fewer the model updates, but a model of a higher order implies more multiplication operations. On the contrary, the lower the order of the model, the lower the precision and the less multiplication operations, but the more model updates. Therefore, in practice, we should choose a reasonable order and error range according to the specific environment and experience.

**Communication cost**

Figure 4 describes how the communication cost varies under different numbers of queries. With the increasing number of queries, the communication costs of all data aggregation schemes except the ARIMA scheme proposed in this article are relatively stable. Since frequent information interactions within clusters are indispensable in the CPDA method, its communication cost is the highest. In contrast, the communication cost of TAG is much less than CPDA, RPDA, and ESDA. The communication cost of ESDA, in which the number of packets transmitted in the network is substantially smaller, is less than that of RPDA.

For ARIMA, in the first few queries, since there is less historical data, the predictions are less accurate. Thus, nodes in the cluster need to update the ARIMA model more frequently and send the new model and recent historical data to the cluster head, leading to a higher communication cost. After that, the model becomes stable. If the model needs to be updated, the node sends a message; otherwise, no communication is necessary. As a result, the communication cost of subsequent queries becomes stable, which is close to that of TAG.

**Accuracy**

Accuracy is an important metric to evaluate the performance of a data aggregation algorithm. Generally, in the course of data transmission, the accuracy of data aggregation cannot reach 100% due to the impact of channel noise, data transmission delay, and data collision. The experimental results show that the accuracy increases with the increase of the query time intervals. In theory, if there is no data loss, the accuracy of each method could reach 100% theoretically. However, in practice, data collisions in wireless channel cause packet losses. As a result, the aggregation accuracy will be affected.

It can be observed from Figure 5 that the larger the query time interval, the better accuracy. This is because the chance of having data collisions is lower with a longer query time interval. ARIMA has the highest overall accuracy. In contrast, CPDA has the lowest because nodes in a cluster need frequent information exchanges, resulting in a high delay. The accuracy of TAG is next to ARIMA, slightly higher than ESDA. It is worth noting that ESDA has a smaller traffic volume. Although ESDA is a prediction-based model, the prediction error will not affect data aggregation because the cluster head will generate the complete original observation value by adding the data sent from each node in the cluster.

Figure 6 gives the relationship between accuracy and the aggregation time of the proposed scheme under 60, 80 and 100 nodes. It indicates that aggregation accuracy decreases with the increasing node density.

**Flexibility**

The current schemes for data aggregation usually support only a single data type and are not suitable for multi-application scenarios. Due to the unlimited number of applications theoretically supported by ARIMA, it is more flexible, where the length of ciphertext is related to the number of applications \( k \). In our solution, the number of applications supported is related to the network scale and the maximum length of plaintext
collected by a single node. Supporting that the maximum length of each data type collected by a single node is 8 bits and the length of ciphertext is constant, the relationship between the number of applications supported $k$ and the number of nodes $n$ is exhibited by Figure 7. When the number of nodes reaches 600, the maximum number of applications allowed in the network can still be 13, and it is indicated that the proposed scheme is flexible.

**Summary and outlook**

In this article, we proposed a secure data aggregation scheme for wireless sensor data based on the ARIMA model. The scheme requires a cluster head to store the prediction model of all nodes in the cluster and all nodes to make predictions synchronously during data aggregation. If the prediction error is less than the predetermined threshold, the prediction value is taken as the aggregated value. When a node needs to update the model, new data and a new ARIMA model will be sent to the cluster head in the form of the prediction error plus a private random number.

This scheme aims to improve prediction accuracy, reduce the number of communications among nodes in a cluster, and ensure data privacy. Based on the Hadoop framework, the proposed secure data aggregation method can effectively analyze and integrate a variety of data generated by a WSN while preserving data privacy. The experimental results show that the ARIMA model outperforms a few competing methods in terms of accuracy, computation cost, and communication cost. Our method also achieves desirable data privacy, which is not provided by any of the competing methods.

While data privacy is important in the process of data aggregation, data integrity is equally important. This article does not consider the protection of data integrity. In future work, we will study how to integrate data integrity into a prediction-based model.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China under grant nos 61370084 and 61872105 and the Fundamental Research Funds for the Central Universities under grant no. 3072019CF0603 and the Project for Innovative Talents of Science and Technology of Harbin under grant no. 2016RAXXJ013.

**ORCID iD**

Hongtao Song https://orcid.org/0000-0001-6132-5843

**References**

1. Rault T, Bouabdallah A and Challal Y. Energy efficiency in wireless sensor networks: a top-down survey. *Comput Netw* 2014; 67: 104–122.
2. Du X, Shayman M and Rozenblit M. Implementation and performance analysis of SNMP on a TLS/TCP base. In: *Proceedings of the 7th IEEE/IFIP international symposium on integrated network management proceedings: integrated management strategies for the new millennium (IM)*, Seattle, WA, 14–18 May 2001, pp.453–466. New York: IEEE.
3. Huang X and Du X. Achieving big data privacy via hybrid cloud. In: *Conference on computer communications workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 27 April–2 May 2014, pp.512–517. New York: IEEE.
4. Li XR and Bar-Shalom Y. Multiple-model estimation with variable structure. *IEEE Trans Autom Contr* 1996; 41(4): 478–493.
17. Ghaddar A, Razafindralambo T, Simplotryl I, et al.

15. Amidi A. ARIMA based value estimation in wireless sensor networks.

14. Lu J, Valois F, Dohler M, et al. Optimized data aggregation of encrypted data in wireless sensor networks.

13. Tulone D and Madden S. PAQ: time series forecasting with ARIMA and fuzzy logic for VANETs.

12. Ramachandran S, Gopi AK, Elumalai GV, et al. REDD: a new approach for approximate query answering in sensor networks.

11. Mantri D, Prasad NR and Prasad R. BHCDA: bandwidth-efficient heterogeneity aware cluster based data aggregation for wireless sensor network.

10. Patil P and Kulkarni U. SVM based data redundancy elimination for data aggregation in wireless sensor networks.

9. Avokh A and Mirjaily G. Dynamic balanced spanning tree (DBST) for data aggregation in wireless sensor networks.

8. Madden S, Franklin MJ, Hellerstein JM, et al. Tag: a data aggregation service for ad-hoc sensor networks.

7. Zhao Y, Ghosh BK and Cheng D. Control of large-scale Boolean networks via network aggregation.

6. Caballero-Gil P, Molina-Gil J and Caballero-Gil C. Data aggregation based on fuzzy logic for VANETs.

5. Galluccio L, Palazzo S and Campbell AT. Efficient data aggregation in wireless sensor networks: an entropy-driven analysis.

4. Lu J, Valois F, Dohler M, et al. Optimized data aggregation in wireless sensor networks.

3. Tulone D and Madden S. PAQ: time series forecasting with ARIMA and fuzzy logic for VANETs.

2. Ramachandran S, Gopi AK, Elumalai GV, et al. REDD: a new approach for approximate query answering in sensor networks.

1. Mantri D, Prasad NR and Prasad R. BHCDA: bandwidth-efficient heterogeneity aware cluster based data aggregation for wireless sensor network. In: *International conference on advances in computing, communications and informatics (ICACCI)*, Mysore, India, 22–25 August 2013, pp.1309–1316. New York: IEEE.

11. Mantri D, Prasad NR and Prasad R. BHCDA: bandwidth-efficient heterogeneity aware cluster based data aggregation for wireless sensor network. In: *International conference on advances in computing, communications and informatics (ICACCI)*, Mysore, India, 22–25 August 2013, pp.1064–1069. New York: IEEE.

12. Ramachandran S, Gopi AK, Elumalai GV, et al. REDD: redundancy eliminated data dissemination in cluster based mobile sinks. In: *International conference on recent trends in information technology (ICRTIT)*, Chennai, India, 3–5 June 2011, pp.330–335. New York: IEEE.

13. Tulone D and Madden S. PAQ: time series forecasting for approximate query answering in sensor networks. In: *European workshop on wireless sensor networks (EWSN)*, Zurich, 13–15 February 2006, pp.21–37. Berlin: Springer.

14. Lu J, Valois F, Dohler M, et al. Optimized data aggregation in WSNs using adaptive ARMA. In: *Proceedings of the 4th international conference on sensor technologies and applications (SENSORCOMM)*, Venice, 18–25 July 2010, pp.115–120. New York: IEEE.

15. Amid A. ARIMA based value estimation in wireless sensor networks. In: *Int Arch Photogram Remote Sens Stat Inform Sci* 2014; 40(2/W3): 41–46.

16. Li G and Wang Y. Automatic ARIMA modeling-based data aggregation scheme in wireless sensor networks. *EURASIP J Wireless Commun Netw* 2013; 2013: 85.

17. Ghaddar A, Razafindralambo T, Simplotryl I, et al. Towards energy-efficient algorithm-based estimation in wireless sensor networks. In: *Proceedings of the 6th international conference on mobile ad-hoc and sensor networks (MSN)*, Hangzhou, China, 20–22 December 2010, pp.39–46. New York: IEEE.

18. Du X, Guizani M, Xiao Y, et al. Defending DoS attacks on broadcast authentication in wireless sensor networks. In: *International conference on communications (ICC)*, Beijing, China, 19–23 May 2008, pp.1653–1657. New York: IEEE.

19. Poornima AS and Amberker BB. SEEDA: secure end-to-end data aggregation in wireless sensor networks. In: *Proceedings of the 7th international conference on wireless and optical communications networks (WOCN)*, Colombo, Sri Lanka, 6–8 September 2010, pp.1–5. New York: IEEE.

20. Rivest RL, Adleman L, Dertouzos ML, et al. On data banks and privacy homomorphisms. *Found Secure Comput* 1978; 4(11): 169–180.

21. Castelluccia C, Mykletun E, Tsdik G, et al. Efficient aggregation of encrypted data in wireless sensor networks.

22. Papadopoulos S, Kiayias A and Papadias D. Secure and efficient in-network processing of exact sum queries. In: *Proceedings of the 27th international conference on data engineering (ICDE)*, Hannover, 11–16 April 2011, pp.517–528. New York: IEEE.

23. Li Z and Gong G. Data aggregation integrity based on homomorphic primitives in sensor networks. In: *Proceedings of the 9th international conference on Ad-Hoc networks and wireless (ADHOC-NOW)*, Edmonton, AB, Canada, 22–28 August 2010, pp.149–162. Berlin: Springer.

24. Zhong H, Shao L, Cui J, et al. An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks. *J Parallel Distrib Comput* 2018; 111: 1–12.

25. Gopikrishnan S and Priakanth P. HSDA: hybrid communication for secure data aggregation in wireless sensor network. *Wireless Netw* 2016; 22(3): 1061–1078.

26. Yager RR. Exponential smoothing with credibility weighted observations. *Inform Sci* 2013; 252: 96–105.

27. He W, Liu X, Nguyen HV, et al. PDA: privacy-preserving data aggregation for information collection. *ACM Trans Sens Netw* 2011; 8(1): 6.

28. Zhang X, Chen H, Wang K, et al. Rotation-based privacy-preserving data aggregation in wireless sensor networks. In: *International conference on communications (ICC)*, Sydney, NSW, Australia, 10–14 June 2014, pp.4184–4189. New York: IEEE.

29. Yu T, Wang X and Shami A. Recursive principal component analysis-based data outlier detection and sensor data aggregation in IoT systems. *IEEE IoT J* 2017; 4(6): 2207–2216.

30. Vinodha D and Anita EAM. Secure data aggregation techniques for wireless sensor networks: a review. *Arch Comput Methods Eng* 2019; 26(4): 1007–1027.

31. Kumar H and Singh PK. Comparison and analysis on artificial intelligence based data aggregation techniques in wireless sensor networks. *Procedia Comput Sci* 2018; 132: 498–506.

32. http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring