Aggregation-tolerant steganography for data-stream integrity verification

A. Soltani Panah and R. van Schyndel

Data outsourcing can make data-integrity protection a challenging task, especially when the trustworthiness of a third-party is unproven. A novel auditing process for integrity verification of data stream, whose storage and handling is outsourced to a third-party, is explored. For this purpose, the hidden information within this data that provides support for in-network data aggregation environments, such as sensor networks is masked. This mask is achieved by simultaneous embedding of several shifted watermark patterns into multiple data streams. The nature of this watermark allows it to be aggregated into a single data stream with minimal loss of this information. A great benefit of the proposed scheme is that the embedded watermarks are invariant to linear time-sequential or cross-stream aggregation operations, such as summation or averaging. Therefore, multiple data streams can be merged and at the same time, and the existence of each individual watermark within allowable bounds can still be verified. The simulation results show that the embedded watermarks can successfully be recovered with high confidence if proper hiding codes are chosen.

Introduction: In the broad classes of monitoring applications such as sensor networks, data is continuously collected from multiple sources and transmitted to clouds after in-network aggregation. One of the biggest concerns with cloud data storage is that of data-integrity verification at untrusted servers. Cloud providers may hide data loss incidents for the sake of their reputation or by discarding rarely accessed data for monetary reasons [1]. Moreover, before outsourcing, data may go through various layers of aggregation, summarisation and compression, often making reasoning about the final result’s origins nearly impossible [2]. To provide an audit trail of the data, we propose a novel technique based on spread spectrum (SS) watermarking. This enables the users to verify the integrity of the outsourced data without having access to a local copy. This is referred to as blind detection.

In our scheme, each data source adds a unique pseudo-random (PN) code to its sensory data periodically, and this is aggregated to a single stream. This aggregator can also potentially embed its own unique PN code. Recovery is based on correlation with a template array for validating aggregated in-band signal. In this Letter, we show the results for Gold codes, but the process is independent of choice of sequence, and we will expand in future work with more diverse and secure sequences.

Contributions: We deal with the question of data provenance – assertions that the data is complete and accurate. We provide a method of in-channel code insertion that travels with the data, and is not metadata such as labels or headers. We observe that such codes can be aggregated without loss of information other than by quantisation, provided the codes used have certain properties – in our case, they must be from a family of PN sequences with good cross-correlation properties. We describe a mechanism of extraction that allows a single correlation to extract all watermarks within an aggregated data stream and its child nodes. We assert that such data can be verified, not only on the data itself, but also to a limited extent on its aggregates. Finally, we assert that the above features allow a data consumer to verify the data, given that it has knowledge of the embedded code values and this data consumer need not be the cloud-provider.

The problem of watermarking a single, discrete stream of data such as sensory data was first addressed in [3]. In many cooperative-sensing environments, sampling and aggregation are common operations. Therefore, we provide a watermarking scheme that could survive these kinds of operations. The advantages of our method are listed as follows:

- **Private data verifiability**: Instead of delegating the task of data evaluation to a trusted third-party, our approach can be used for verifying the trustworthiness of data at the cloud end, by the entity/person that knows the embedded values, which need not be the cloud-provider.
- **Statistical invisibility**: The use of SS codes affords us most of the desirable statistical properties of such codes. The properties being a resemblance to additive white Gaussian noise, near-ideal auto-correlation and near-ideal cross-correlation with other family members. This makes the watermark more robust against linear statistical tools.
- **Aggregation invariance property**: The embedded watermarks survive common linear aggregation operations due to the correlation properties of the codes used.
- **Watermark localisation**: The aggregation-tolerant property of the scheme further benefits our design for locating missing or altered watermarks by finding the corresponding missing correlation peaks. This is achieved because of embedding unique shifted versions of different PN codes of a family.
- **Limited scale invariance**: Only one correlation is sufficient to extract all the watermarks, due to the family property, giving \( O(1) \) scaling for a number of watermarks. The number of watermarks is primarily limited by data resolution of each channel.

Watermark generation and embedding: We consider a sensor network consisting of a set of \( N \) nodes with a two-tier hierarchy, including leaf nodes and aggregators such that sensory data from leaf nodes are forwarded to the aggregators. Assume a family of binary PN sequences of length \( l \) with good correlation properties denoted as \( S_f = \{s_1, s_2, ..., s_p\} \) with \( p \) members \( (p \geq N) \). We assign a unique pair \( (s_i, \lambda_i) \) to each individual node \( n_i \), including a unique family member \( s_i \) and a unique integer value as shift, that is, \( s_i \in \{0, ..., T - 1\} \), where \( T = m \times l \) and \( m \) is an integer chip-rate, so we have a sampling window with length equal to an integer number of sequences. The essence of watermark embedding is adding the scaled version of the shifted PN sequence to the data as follows:

\[
d^w(t) = d(t) + g \times s_i^n(t)
\]

where \( d(t) \) is the sample of data stream at time \( t \) belonging to node \( n_i \) and \( g \) corresponds to the global scaling factor set to render the watermark imperceptible from the noise floor of the original sensor data. The symbol \( s_i^n(t) \) is the shifted PN sequence \( s_i \) at time \( t \) such that \( s_i^n = \text{rot}(s_i, \lambda_i) \) where \( \text{rot}() \) represents a spatial shift of \( s_i \) by \( \lambda_i \) with cyclic wrap around. By decoupling the sample window size \( T \) from the sequence length \( l \), we enable sampling over an unbounded stream that slides continuously over the length of the window. Therefore, where finite (aperiodic) data is involved, the watermark is replicated to resemble periodic boundary conditions affording the use of periodic correlation. The aggregator(s) in turn aggregates the result from its children, possibly embedding its own unique watermark, and forwards the aggregated data stream to the parent node in the path. We rewrite the summation of (1) as a linear aggregation as follows:

\[
d^w_{agg}(t) = s_{agg}(t) + \sum_{i \in N_{agg}} d_i(t), \quad N_{agg} \subseteq N
\]

where \( s_{agg}(t) \) represents the sum of all scaled watermarks for that aggregator.

The final aggregator might forward the data stream to an analyser for the watermark detection process. We assume that at the beginning of an aggregation round, all sensor clocks are synchronised; otherwise the watermark detection process needs to search over a large space to find the whole pattern which increases the detection complexity. A feature of this method is the simplicity of implementation at the sensor end.

Watermark detection: For the decoding process, we use a full periodic cross-correlation, which requires the examination of at least one full sequence length \( l \) of data. A periodic correlation is used to retrieve significant peaks at the corresponding shifts \( \lambda \). The detection is not completely error-free, owing to partial cross-correlation of the aggregated data with the encoded sequences. Hence, before performing the correlation, a few steps are necessary in order to boost the detection performance. The detection process involves the following steps:

- Remove the mean value of the received signal (aggregated watermarked data series).
- Compute the correlation between the received signal after removing the mean value \( r(t) \) and a template array \( o(t) \), which is the summation of all assigned spreading codes without any rotation over the sequence length as \( R(t) = \sum_{\tau \in T} o(\tau + t) \).
- Apply a matched filter that passes the correlation peaks but suppresses non-impulsive noise.
- Apply a peak detection algorithm for finding correlation peaks.
The first step is necessary for minimising the interference due to multiple data streams because PN sequences have almost zero mean (balance property of PN sequences) and so should the data. Additionally, in the third step, we apply a matched filter after correlation in order to reduce the interference with other sequences and residual data. If the chip-rate is one, a simple differential filter should be sufficient as a matched filter since the auto-correlation peaks of sequences are impulsive and can survive the filtering. After these steps, an impulsive peak detection algorithm can be used for finding significant correlation peaks and their positions.

If detection is successful, the number of retrieved correlation peaks over window size \( T \) is equal to the number of nodes \( n \), and the absolute peak position for each node \( n_i \) is equivalent to its unique shift \( \lambda_i + 1 \). One may argue that since each node has a unique PN code, there is no need for using shifts. The rationale behind assigning unique shifts is that, by doing so, the detection process only requires one correlation in order to find correlation peaks. Otherwise, we need to perform cross-correlation against each of the assigned PN codes to determine whether a node contributes to the aggregated data flow. This is not desirable in a large network because of the overhead of cross-correlation calculations. An example of our solution is illustrated in Fig. 1. There are nine nodes in the network, including leaf nodes (white) and aggregators (black). We used a unique rotated member of Gold codes of length 511, embedded by each node in its sensory data. The existence of correlation peaks and the correct location of those peaks indicate that the nodes are working properly.

**Applications:** As a primary motivation, we consider environmental monitoring applications, where the main objectives are to understand the sensing field based on the aggregation of all sensory data, such as temperature distribution or air pollution. Similarly, in supervisory control and data acquisition networks, the highest cost components of systems are continuously monitored in order to detect failures before they reach a catastrophic stage. In these applications, if each data source embeds its unique watermark, then faults could be detected by looking for the corresponding watermarks that are missing from the aggregated data stream. If each data source along the path inserts its unique watermark, this chain of aggregated watermarks could be considered as data provenance, which shows the tracking path along with the data [4] and is crucial for assurance of data trustworthiness.

**Simulation results:** To evaluate the feasibility of our scheme, the publicly available sensory data from a small SensorScope network [5] was chosen. It contained humidity measurements collected every 30 s by 23 weather stations. As the samples had enough data resolution (12 bits analogue-to-digital converter), the global scaling factor \( g \) was set to 1. We assigned each of 23 nodes a unique member of the Gold codes and equally spaced shifts \( \lambda_i = i \times (T/23) \). This results in maximum distance between adjacent correlation peaks and therefore decreases the probability of false detection that might happen by time-offset misalignment. To evaluate the scheme’s performance under data modification, an alteration attack [3] is modelled, where a percentage of the samples are randomly modified with a uniform noise distribution with \( (\varepsilon = 0.6, \mu = 0) \) as the amplitude and the mean of alteration for stations with ‘even’ ids (11 stations in total). Then watermarked data from all the stations were added together to simulate data aggregation. The watermark detection probability as a function of sequence length is presented in Fig. 2. The results indicate that using a family of longer sequences improves the detection probability, as expected, and for the sequence length of 1023, when the percentage of alteration reaches roughly \( (3\% \times 11 = 33\%) \) of the aggregated data, the modification can be recognised with 97% confidence, which means that higher data distortion results in more accurate detection.

**Conclusion:** In this Letter, we proposed a watermark-based solution for the verification of outsourced data which can robustly cope with linear aggregation of multiple data streams. The provided scheme is unique in the way that it exploits the correlation properties of the signature codes used. Moreover, the embedded patterns are balanced, and the linear statistical properties of the data such as average are unaltered. For a small sequence family with near-perfect correlations properties, the number of nodes is limited. However, there are a number of possible ways for increasing this, such as using clustering networks and assigning signature codes to each cluster (head) rather than each node, or applying higher dimensional construction such as in [6] for increasing family size. The correlative decoding process requires the examination of full sequence length of data and must be performed offline. Alternatively, a shorter sequence length can be used to decrease the processing time, but it decreases the detection probability according to the simulation result. Therefore, there exists a trade-off between the desired error granularity and the detection time which leads to different design decisions that might be adapted.

© The Institution of Engineering and Technology 2014
10 July 2014
doi: 10.1049/el.2014.2493
One or more of the Figures in this Letter are available in colour online.

A. Soltani Panah and R. van Schyndel (School of Computer Science and IT, RMIT University, Melbourne 3000, Australia)
E-mail: arezou.soltanipanah@rmit.edu.au

**References**