CONFIDENTIALITY AND INTEGRITY FOR SUM AGGREGATION IN SENSOR NETWORKS

Keith B. Frikken and Yihua Zhang
Miami University, Oxford, Ohio, U.S.A.

Keywords: Sensor aggragation, Privacy, Integrity.

Abstract: When deploying sensor networks in environments that monitor people (e.g., monitoring water usage), both privacy and integrity are important. Several solutions have been proposed for privacy (Castelluccia et al., 2005), (He et al., 2007), and integrity (Yang et al., 2006), (Przydatek et al., 2003), (Hu and Evans, 2003), (Chan et al., 2006), (Frikken and Dougherty, 2008). Unfortunately, these mechanisms are not easily composable. In this paper, we extend the splitting schemes proposed in (He et al., 2007) to provide privacy and integrity when computing the SUM aggregate. Our scheme provides privacy even if the base station colludes with some cluster heads, and provides integrity by detecting when individual nodes inflate or deflate their values too much. Our main contributions are: i) a new integrity measure that is a relaxation of the one in (Chan et al., 2006), ii) a new privacy measure called *k*-similarity, iii) a construction that satisfies both of these measures for the computation of the SUM aggregate that avoids the usage of expensive cryptography, and iv) experimental results that demonstrate the effectiveness of our techniques.

1 INTRODUCTION

Wireless sensor networks have promising applications from military surveillance to civilian usage. In these applications, the base station queries the network and sensor nodes report their values to the base station. In some applications, privacy and integrity are security concerns. For example, if the sensor's individual readings reveal information about specific people, then these values must be protected (even against the base station). Furthermore, as individual nodes may become compromised the base station desires a guarantee about the accuracy of the query result.

A well known technique to extend the lifetime of the network is in-network aggregation. Although this approach reduces the communication overhead and extends the network's operation time, in its most straight-forward implementation it suffers from both privacy and integrity problems (for a detailed survey of security in aggregation see (Alzaid et al., 2008)). In terms of privacy, while the base station only receives an aggregated result, the values are now leaked to other nodes(i.e., aggregator nodes) in the network. Also, there are now two integrity threats: i) a node may inflate of deflate its values (and the base station can no longer perform the sanity check on each value) and ii) an aggregator might misrepresent the aggregated value. There has been a significant amount of work (Castelluccia et al., 2005) and (He et al., 2007) addressing the privacy issue for in-network aggregation. However, all of these works assume that sensors will honestly report their values. Also, many schemes (Yang et al., 2006), (Przydatek et al., 2003), (Hu and Evans, 2003), (Chan et al., 2006), (Frikken and Dougherty, 2008) address the integrity issue. Basically, these schemes use other nodes to verify the validity of reported values. However, with these approaches a verifying node inevitably learns the sensitive information for the nodes that it verifies. The natural question becomes "Can we design a scheme that achieves both privacy and integrity?" The only work that we are aware of that addresses both of these problems is (Roberto et al., 2009) and (Castelluccia and Soriente, 2008). We discuss the differences between our work and this prior work in the next section.

We propose a scheme for computing the sum aggregate that provably achieves both meaningful privacy and integrity. Our work is built upon the SMART scheme(He et al., 2007), that uses the split-and-merge mechanism. Our main contributions are:

1. We introduce the notion of amplification factor to measure the deviation degree between the re-

B. Frikken K. and Zhang Y. (2010).
CONFIDENTIALITY AND INTEGRITY FOR SUM AGGREGATION IN SENSOR NETWORKS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 231-240
DOI: 10.5220/0002937602310240

ported and the correct aggregate values.

2. We introduce a new privacy notion, *k*-similarity, that provides "good" enough security. We provide analysis to show that this new notion provides a reasonable level of privacy.
3. We provide a protocol for computing the sum aggregate that achieves both integrity and privacy. The proofs of these claims is omitted due to page constraints. Expensive cryptography is not used in this protocol, which makes it applicable to current sensor technology. Furthermore, the communication is also reasonable.
4. We provide experimental results to demonstrate the effectiveness of our approach.

The rest of this paper is organized as follows: in section 2 we survey related literature. In section 3 we define the problem, and in section 4 we introduce a splitting scheme that performs the sum aggregation. In section 5, we formally define the integrity and privacy goals for splitting schemes. In section 6, we provide a construction that satisfies our goals. In section 7, a series of experiments to test the effectiveness of our constructions is performed. Finally in section 8, we summarize our work and describe future work.

2 RELATED WORK

Initial works(Madden et al., 2002), (Intanagonwiwat et al., 2002) in the data aggregation domain share the same assumption that all sensors in the network are honest, and no outsiders attempt to eavesdrop or tamper with the sensor readings. However, in reality, sensors are deployed in unattended or hostile environments which put them at risk in the following ways: i) adversaries interested in the values of individual sensors will either eavesdrop the communication or physically crack the sensors to obtain the sensor readings and ii) adversaries who compromise a fraction of sensors will attempt to mislead the base station to accept a spurious aggregate result or prevent the final aggregate from being reported to the base station. Many schemes have been introduced that address either the privacy or the integrity issue.

Many of the approaches for integrity utilize a divide-and-conquer, commit-and-attest mechanism for the purpose of obtaining an acceptable aggregation result when a fraction of sensors are compromised. These schemes fail to achieve privacy, because in the aggregation phase, the intermediate node(aggregator) will learn all private sensor readings sent from its children. In (Chan et al., 2006) and (Frikken and Dougherty, 2008), schemes for provably secure hierarchical in-network data aggregation were proposed. The schemes were based on a commit-and-attest mechanism, but utilized the delayed aggregation to effectively reduce the verification overhead. However, neither of these schemes provided privacy.

Works presented in (Castelluccia et al., 2005), (He et al., 2007) are related to privacy preservation. In (Castelluccia et al., 2005), the author proposed a homomorphic encryption scheme that achieves both the end-to-end privacy and energy-efficient properties. Another work (He et al., 2007) introduced two schemes(CPDA, SMART) to protect individual sensor readings during the aggregation phase. Specifically, in the SMART scheme each sensor conceals its private data by slicing it into pieces, and sends the pieces to different cluster heads in the networks. After receiving all shares, those cluster heads will simply aggregate those shares, and further send the aggregate to the base station. One major problem in these two works(Castelluccia et al., 2005), (He et al., 2007) is that they did not consider the existence of malicious users who may report illegal values (e.g., values that are outside of the range of legal values). Lacking a mechanism to check the validity of reported values, the integrity will not be guaranteed. Our work is primarily based on analyzing the security characteristics of the SMART scheme, and aims to incorporate both the confidentiality and integrity into this scheme. Source location privacy(Kamat et al., 2005), (Yang et al., 2008) attempts to hide the location of a reported event, which is an orthogonal issue to the issues considered in this paper.

Recently, a scheme(Roberto et al., 2009) was proposed to address both of the privacy and integrity issues. In (Roberto et al., 2009), the author applies homomorphic encryption to preserve the privacy, and uses monitoring sensors to detect the abnormal behavior of aggregators. That is, each aggregating node has several monitoring nodes that ensure that the aggregator does not misbehave. However, this does not prevent leaf nodes from intentionally reporting illegal values. In order to preserve privacy, the scheme also requires that neither the aggregating nodes nor the monitoring nodes collude with the base station.

Another scheme, ABBA, (Castelluccia and Soriente, 2008) was proposed for providing privacy and integrity in sensor aggregation which utilized an additive checksum to provide integrity. A downside with this approach was that if an adversary corrupted a single node and knows the reported values of several nodes in the network, then this adversary can modify these known values to arbitrary values.

3 PROBLEM DEFINITION

We consider a sensor network with N nodes, denoted by $s_1, \ldots, s_N$. At any given time each sensor node has a value in the range $[0, M]$, where node s_i's value is denoted by v_i. A special node, the base station, will query the network to learn $\sum_{i=1}^{N} v_i$. Limiting the range of values to $[0, M]$ does not limit the applicability of the scheme, because other ranges can simply be scaled to match this type of range.

We assume that there is a special set of nodes, called cluster heads. These cluster heads could either be more expensive tamper-resistant nodes, or can be regular nodes in the network that are identified via a cluster formation process (as in (He et al., 2007)). During deployment (before the adversary compromises nodes), each sensor node discovers its closest C cluster heads. The regular nodes in the network will send information to their closest cluster heads, and then these cluster heads will pass the aggregation information up the aggregation tree to the base station.

We assume that sensor nodes have keys with each other. Specifically, we require that each sensor node has a key with each of its C cluster heads. This can be achieved using one of the many well-known key predistribution schemes (see (Camtepe and Yener, 2005) for a survey of such schemes). We assume that the base station can perform authenticated broadcast by using a protocol such as μTELSA(Perrig et al., 2002).

The primary security concerns in this paper are:

1. **Integrity.** We want to defend against the stealthy attack defined in (Przydatek et al., 2003). Specifically, we assume that some leaf nodes are compromised, and we want to prevent them from convincing the base station of a false result. At a high level, we want to prevent nodes from being able to have more influence on the final aggregate than what can be achieved by changing its reported value to something in the range $[0, M]$. We achieve only a weakened form of this goal. Essentially, we bound the amount that each node cannot influence the result. When using in-network aggregation, there are two potential threats: i) a sensing node is corrupt and reports a value outside of the range $[0, M]$, and ii) aggregating node modifies the results of previous nodes. The scheme in (Chan et al., 2006) protected against both of these threats, but did not preserve privacy. The work in (Roberto et al., 2009) did not consider the first type of attack, and thus to corrupt the result, an adversary only needs corrupt an individual node. In this paper we focus on preventing this type of attack, but do not address corrupt aggregating nodes. However, our techniques can be combined with the technique in (Roberto et al., 2009) to protect against both corrupt reporting nodes and corrupt aggregating nodes.
2. **Privacy.** The value of each sensor node should be private, even from the base station. We assume that the base station and up to t cluster heads are corrupt, and attempt to learn an individual sensor's values. Here "corrupt" means that the cluster heads collude with the base station to reveal the sensor's private data, not that it lies to the base station(i.e., misrepresent the sensor's private reading, or not warn the base station about illegal shares). We prove security when $t = 1$, but provide experimental results that show that our approach is effective for larger t values.
3. **Availability.** We do not consider denial of service attacks in this paper.

4 SUM AGGREGATION USING SPLITTING SCHEMES

In this section, we will review the basic ideas of the SMART (He et al., 2007) splitting scheme that aims to preserve the privacy of each sensor's reading during the sum aggregation. Each sensor hides its private data by slicing it into several pieces, and then it sends each encrypted piece to different cluster heads. After receiving all pieces, each cluster head will calculate the intermediate aggregate result, and further report it to the base station. To explain in details, we will divide this process into three steps: Slice, Mix and Merge.

Step 1 ("Slice"). Each node s_i, will slice v_i into C shares: $v_i^1, \ldots, v_i^C$. That is $v_i = \sum_{j=1}^{C} v_i^j$. The node then sends to each of its C cluster heads one of these values.

Step 2 ("Mix"). After receiving all of the shares, the cluster head decrypts all of its values and sums up all of the reported shares. It then sends this aggregate to the base station.

Step 3 ("Merge"). The base station receives all of the values from the cluster heads and sums up all of these values to obtain the sum of all nodes' values. This value will be $\sum_{i=1}^{N} \sum_{j=1}^{C} v_i^j = \sum_{i=1}^{N} v_i$.

It is important to note that the actual SMART protocol is slightly different from the one above. Specifically, each node would sends their shares to a random subset of nodes in the network. Also, the nodes keep one share for themselves, aggregate it with other shares it received. While the SMART protocol intuitively achieves private aggregation for the SUM, there are two main limitations to its initial presentation in (He et al., 2007). First, no description of how to split the values was given in (He et al., 2007), and second, no formal analysis was given to provide any security guarantees. However, this approach does potentially enjoy an additional advantage that was not discussed in (He et al., 2007); specifically, it can potentially provide integrity in addition to privacy. That is, in the mixing step, the cluster heads can verify that the values are in a valid range; and thus, this could bound the amount a corrupted node can affect the final aggregate. The aims of this paper are to: i) give a specific construction for splitting schemes, ii) provide a formal analysis of the effectiveness of this scheme with regards to privacy and integrity, and iii) demonstrate that the construction provides a meaningful notion of privacy and integrity.

5 FORMALIZING SPLITTING SCHEMES

In this section we formalize the desired properties of a splitting scheme, so that it can be used in a protocol such as the one in the previous section to provide integrity and privacy. Suppose a sensor node has a value v that is in the range $[0,M]$. The node is concerned about the privacy of v so it splits v into integer shares $v_1,\ldots,v_s$ such that $\sum_{i=1}^{s} v_i = v$. The node then reports to each of its cluster heads one of these values. The cluster heads then verify that each share is in a valid range and then aggregate the individual shares. The privacy concern with this approach is that an adversary would obtain some share values (i.e., by corrupting some cluster heads) and then be able to determine information about v. Informally, the goal is that if the adversary obtains up to some threshold t shares, the adversary should not be able to determine the value used to generate the shares. In the remainder of this section we focus on the case where $t=1$, but in section 7 we consider larger values of t. What complicates this problem, is the orthogonal server goal of data integrity; that is the server wants to prevent the client from inflating or deflating his value too much (i.e., reporting a value outside of the range $[0,M]$. We now formalize some notions about splitting values.

Definition 1. *A splitting scheme is a probabilistic algorithm S that takes as input: i) an upper bound on values M, ii) a value* $v\in[0,M]$*, and iii) a number of shares s. S then produces output* $v_1,\ldots,v_s$ *such that* $\sum_{i=1}^{s} v_i = v$.

To simplify the analysis of splitting schemes we will consider only splitting schemes where the distribution for share v_i is the same as the distribution for the share v_j for all $i,j\in[1,s]$. More formally:

Definition 2. *A splitting scheme, S, is called* symmetric *if* $\forall i_1,i_2\in[1,s]$ *and* $j\in\mathbb{Z}, Pr[v_{i_1}=j|(v_1,\ldots,v_s)\leftarrow S(M,v,s)] = Pr[v_{i_2}=j|(v_1,\ldots,v_s)\leftarrow S(M,v,s)]$ *for all valid choices of M, v, and s.*

One may be concerned that asymmetric splitting schemes may perform better than symmetric splitting schemes. However, suppose we have an asymmetric splitting scheme, A. To convert A into a symmetric scheme: i) compute $(v_1,\ldots,v_s)\leftarrow A(M,v,s)$ and ii) randomly permute the shares to obtain the respective shares. It is straightforward to show that this a symmetric scheme, and clearly if no t shares reveal anything about v in the original set, then no t shares reveal anything about v in the permuted set. Thus it sufficient to consider only symmetric splitting schemes.

5.1 Integrity Goal

The server's integrity concern is that a corrupted sensor may report a value not in the range $[0,M]$. Any splitting scheme will produce shares inside of a specific range, we call the range $[Min,Max]$. Formally,

Definition 3. *The range of a splitting scheme, a share count s and an upper bound M is denoted by range(S,M,s) is* $[Min(S,M,s),Max(S,M,s)]$ *where*

- *$Min(S,M,s)$ (resp. $Max(S,M,s)$) is the minimum (resp. maximum) share value produced by $S(M,v,s)$ over all possible $v\in[0,M]$ and all possible choices for the randomness for S.*

Since individual cluster heads know the value *range(S,M,s)*, they can verify that the shares that it receives are inside this range. If any of the shares are outside this range, then the cluster head reports the error to the base station. Thus a node can cannot report any value outside of the range $[s\times Min(S,M,s), s\times Max(S,M,s)]$ to the base station. The additional range reporting capability is defined in the following metric:

Definition 4. *The amplification factor of a splitting scheme S for parameters M,s is*

$$\frac{s\times Max(S,M,s)-s\times Min(S,M,s)+1}{M+1}$$

As an example, suppose that the user's values must be in the range $[0,2]$, and that the splitting scheme produces two shares each in the range $[-2,2]$. Thus a malicious node can report any value in the range $[-4,4]$, and so the amplification factor is 3, that is a malicious user can report a value in a range that is three times bigger than the range of the actual values. That is, if the splitting scheme has an amplification factor of a, then compromising a single node is like compromising a nodes in a scheme where all reported values are to be in the range $[0,M]$.

Clearly, the goal is to make the amplification factor as close to 1 as possible. In fact, in the absence of the privacy goal, this is trivial. Simply have $s=1$, and have $S(M,v,s)=v$. However, while this provides amplification factor of 1, it clearly provides no privacy.

5.2 Privacy Goal

The initial privacy goal is that the adversary should not be able to determine the value of the result, when given one of the shares. Since the scheme is symmetric we only consider giving the first share to the adversary (that is the first share has the same distribution as every other share, so the adversary will not gain more information from receiving a different share). We formalize this notion for symmetric splitting schemes in the following experiment:

Definition 5. Share Indistinguishability Experiment $Exp_A^{M,s,S}(k)$**.**

1. *A is given the security parameter* 1^k*, the values* M *and* s*. A chooses two values* m_0 *and* m_1 *(both in* $[0,M]$*).*
2. *A bit b is randomly chosen.* $(v_1,\ldots,v_s) \leftarrow S(M,m_b,s)$*. A is given* v_1*.*
3. *A outputs a bit* b'*.*
4. *If* $b=b'$*, then output 1. Otherwise output 0.*

We denote the advantage of A as $Adv_A^{M,s,S}(k) := Pr[Exp_A^{M,s,S}(k)=1]-\frac{1}{2}$. We say that a splitting scheme is cryptographically private if for all probabilistic polynomial time (PPT) algorithms A, $Adv_A^{M,s,S}(k)$ is negligible in k. Here, “negligible” has the standard cryptographic definition. That is, a function $f(k)$ is negligible if for all polynomials P and large enough N: $\forall n>N: f(n)<\frac{1}{P(n)}$.

If we ignore the integrity goal, then it is straightforward to achieve cryptographic privacy. Essentially, the algorithm chooses v_1 uniformly from $[0,M*2^k]$ and sets $v_2=v-v_1$. We omit a formal proof that this is cryptographically private, but the basic idea is that if v_1 is chosen in the range $[M,(2^k-1)M]$, then no information is leaked about v, by either of the individual shares. v_1 is not chosen in this range with probability $(2M)/M2^k=\frac{1}{2^{k-1}}$, which is negligible in k. Unfortunately, this scheme has an amplification factor of 2^{k+1}, which clearly provides no meaningful integrity.

5.2.1 Good Enough Privacy

As can be seen from the previous two sections, it is possible to achieve either integrity or cryptographic privacy for splitting schemes. The natural question that arises is whether it is possible to achieve both simultaneously. Unfortunately, we later show that any splitting scheme with cryptographic privacy has super-polynomial (in terms of the security parameter) amplification factor. This implies that one of the two constraints must be weakened. In order for a splitting scheme to provide any integrity, the amplification factor must be kept small, and thus the question becomes: “Is there a meaningful notion of privacy that can be obtained that suffers only moderate amplification factor?” In the remainder of this section we first prove the impossibility result, describe some failed attempts at privacy, and then introduce and analyse a new notion of privacy called k-similarity.

5.2.2 Impossibility Result

We now show that a symmetric splitting scheme cannot achieve both cryptographic privacy and polynomially-bounded amplification factor. Due to page constraints we omit the formal proof and give only a sketch of the results below. Specifically, the main result is as follows:

Theorem 1. *Any symmetric splitting scheme, S, with parameters M, and s* $(s>1)$ *such that* $Adv_A^{M,s,S}(k)\le\varepsilon$ *has an amplification factor* $\ge\frac{\sqrt{Ms}}{2(M+1)\sqrt{\varepsilon}}$

A consequence of the above theorem is that if the adversary advantage is negligible in k, then the amplification factor must be super-polynomial. That is, M and s are fixed constants, and the ε is in the denominator, so the amplification factor is inversely proportional to the square root of the adversary advantage.

Before sketching the proof, we define the following notation:

Definition 6. *A symmetric splitting scheme, S for range* $[0,M]$ *and shares s induces a distribution on* $[Min(S,M,s),\ Max(S,M,s)]$*for each value* $v\in[0,M]$*. We denote the probability that a share is i given a split value v as:*

$$D_v^{S,M,s}[i]=Pr[v_1=i|(v_1,\ldots,v_s)\leftarrow S(M,v,s)].$$

Theorem 1 rests on the following two lemmas:

Lemma 1. $\sum_{i=Min(S,M,s)}^{Max(S,M,s)} i*D_v^{S,M,s}[i]=\frac{v}{s}$.

Lemma 2. *Suppose* $\exists i\in[Min(S,M,s),Max(S,M,s)]$ *and* $m_0,m_1\in[0,M]$ *such that* $|D_{m_0}^{S,M,s}[i]-D_{m_1}^{S,M,s}[i]|\ge\varepsilon$*, then there exists a PPT adversary A such that* $Adv_A^{M,s,S}(k)\ge\frac{\varepsilon}{4}$*.*

The proof proceeds as follows, using lemma 1 it is possible to show that when splitting two different values that there is a specific value where the value is above $\frac{M}{(r-1)s}$ (where $r = Max(S,M,s) - Min(S,M,s) + 1$) . However, combining this with Lemma 2 this is enough to show unless r is super-polynomial, that the $Adv_A^{M,s,S}(k)$ is non-negligible.

5.2.3 A Definition for Privacy

Clearly, in order for splitting schemes to be useful, we need to relax one of our two goals. If we require cryptographic privacy, then the amplification factor will be large and no useful integrity will be provided. Thus we explore weaker definitions of privacy; our fundamental goal is to place some bound on the information gained by an adversary. Before describing the definition, we look at some failed attempts:

1. We could require that $|D_{m_0}^{S,M,s}[i] - D_{m_1}^{S,M,s}[i]|$ is below some threshold ε for all values m_0, m_1, and i. However, this does not prevent the following situation: $D_{m_0}^{S,M,s}[i] = 0$ and $D_{m_1}^{S,M,s}[i]| \neq 0$. If this situation happens, and the adversary is attempting to distinguish between m_0 and m_1 and the corrupted sample is i, then the adversary is certain that the split value is m_1. Thus the adversary's knowledge gain is potentially arbitrarily large.

2. A stronger condition is that $\sum_{i=Min(S,M,s)}^{Max(S,M,s)} |D_{m_0}^{S,M,s}[i]$ $-D_{m_1}^{S,M,s}[i]|$ is below some threshold ε. However, this has the same problem as the prior approach.

When considering the amount of knowledge gained from a share, an important factor is that the difference between the two distributions at any point is small relative to the values at those points. Using this as motivation we propose the following definition.

Definition 7. *A symmetric splitting scheme, S, with parameters M, and s provides k-similar privacy if and only if* $\forall m_0, m_1 \in [0,M], \forall i \in [Min(S,M,s), Max(S,M,s)]$ *either*

i) $D_{m_0}^{S,M,s}[i] = 0$ *and* $D_{m_1}^{S,M,s}[i] = 0$ *or*

ii) $\frac{\min\{D_{m_0}^{S,M,s}[i], D_{m_1}^{S,M,s}[i]\}}{\max\{D_{m_0}^{S,M,s}[i], D_{m_1}^{S,M,s}[i]\} - \min\{D_{m_0}^{S,M,s}[i], D_{m_1}^{S,M,s}[i]\}} \geq k$

5.2.4 Analysis of k-similar Privacy

We are interested in bounding the "information gain" that the adversary has from the captured share. To model this, suppose that the adversary is trying to distinguish whether the user has a value m_0 or a value m_1. Furthermore, the adversary has some background knowledge regarding the likelihood that the value is m_0, and we represent this as probability P. We stress that we do not assume knowledge of value P, but rather we wish bound the information gain for any value of P. Denote as P^i the adversary's probability that the value is m_0 after seeing a single sample with value i. Below, a bound is placed upon the value $|P^i - P|$, which is independent of i.

Theorem 2. *If a splitting scheme satisfies k-similarity, then* $|P^i - P| \leq \frac{Q-Q^2}{Q+k}$ *where* $Q = \sqrt{k^2+k} - k$

Proof. For the sake of brevity denote as $p_i = D_{m_0}^{S,M,s}[i]$ and $q_i = D_{m_1}^{S,M,s}[i]$. Now, $P^i = \frac{P \times p_i}{P \times p_i + (1-P) \times q_i}$. We consider two cases: i) $p_i \geq q_i$ and ii) $p_i < q_i$.

Case 1. $p_i \geq q_i$**:** It is straightforward to show that $P^i \geq P$. Since $\frac{q_i}{p_i - q_i} \geq k$, $q_i \geq \frac{kp_i}{k+1}$. Now, $P^i = \frac{Pp_i}{Pp_i + (1-P)q_i}$, and this value is maximized when q_i is minimized. Thus, $P^i \leq \frac{Pp_i}{Pp_i + (1-P)\frac{kp_i}{k+1}} = \frac{P}{P + (1-P)\frac{k}{k+1}} = \frac{Pk+P}{Pk+P+(k-kP)} = \frac{Pk+P}{P+k}$. Now $P^i - P \leq \frac{Pk+P}{P+k} - P = \frac{P-P^2}{P+k}$. It is straightforward to show that this value is maximized when $P = \sqrt{k^2+k} - k$.

Case 2. $p_i < q_i$**:** A symmetrical argument can be made to case 1. □

In Figure 1 we plot the the maximal knowledge gain (i.e., $|P^i - P|$) for several value of k. Observe that, as k increases this value decreases rapidly. For example, if a scheme satisfies 7-similarity, then a single sample changes the adversary's belief about the reported value by at most 3.4%, and if the splitting scheme satisfies 10-similarity, the maximum change is 2.4%. Clearly, this is not as strong as the notion of cryptographic privacy, but it may be enough security in some situations.

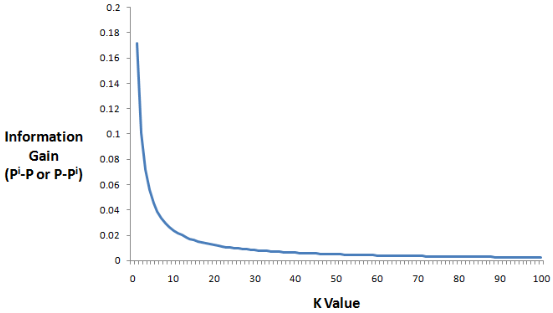

Figure 1: Relation between maximum information gain and K value.

6 A CONSTRUCTION

In this section we present a construction for a splitting scheme. Before describing the scheme we introduce a new definition. Define $C_s(T,a,b)$ to be the number of ways to choose s values that sum up to T where all values are in the range $[a,b]$. Note that these C values can be computed using the following recurrence:

1. $C_1(T,a,b) = 1$ if $T \in [a,b]$ and is 0 otherwise.
2. $C_i(T,a,b) = \sum_{j=a}^{b} C_{i-1}(T-j,a,b)$

The main construction is as follows: At a high level, to split a value v among s shares, the splitting scheme takes as a parameter a value N and produces values in the range $[-N,N]$. We discuss how to choose N later, but clearly it is required that $Ns \geq M$. The scheme chooses q as the first share with probability $\frac{C_{s-1}(v-q,-N,N)}{C_s(v,-N,N)}$. It then chooses shares for values $v-q$ using $s-1$ shares recursively using the same strategy. Before describing the actual construction, we need another building block that chooses a value in a range with the above-specified distribution. This algorithm is given in Algorithm 1, and the details of the construction are provided in Algorithm 2.

Algorithm 1: $CHOOSE(MIN,MAX,v,s)$.

```
for $i = MIN$ to $MAX$ do
    $d_i = \frac{C_{s-1}(v-i,MIN,MAX)}{C_s(v,MIN,MAX)}$
end for
Choose a value $i \in [MIN,MAX]$ according to distribution $d_{MIN},\ldots,d_{MAX}$
return $i$
```

Algorithm 2: $SPLIT(M,v,s,N)$.

```
if $s = 1$ then
    if $v \in [-N,N]$ then
        return $< v >$
    else
        return $FAIL$/*This will never happen*/
    end if
end if
$v_s = CHOOSE(-N,N,v,s)$
$< v_1,\ldots,v_{s-1} >= SPLIT(M,v-v_s,s-1)$
return $< v_1,\ldots,v_s >$
```

In Algorithm 2, notice that if the last share does not fall into the range $[-N,N]$, then the algorithm will return FAIL. However, this situation will never happen, because when we set the ith share to v_i it must be possible to obtain $v-\sum_{j=1}^{i} v_j$ using the remaining shares. This follows from the probability of that the ith share is v_i is: $\frac{C_{s-i}(v-\sum_{j=1}^{i} v_j,-N,N)}{C_{s-i+1}(v-\sum_{j=1}^{i-1} v_j,-N,N)}$, which is 0 if it is not possible to obtain $v-\sum_{j=1}^{i} v_j$ with the remaining shares.

Next, let's look at an example of calculating the share distributions when $N=2$, $M=1$, and $s=3$. We use d_i^j to represent the distribution of share i when splitting the value j. Then, based on the construction, when splitting value $v=0$, we need to follow the formula below to calculate share i's distribution:

$$d_i^0 = \frac{C_2(0-i,-2,2)}{C_3(0,-2,2)}$$

Concerning d^0_{-1}, for the numerator $C_2(1,-2,2)$, since there are four ways($\langle 0,1\rangle,\langle 1,0\rangle,\langle 2,-1\rangle,\langle -1,2\rangle$) to sum up to 1 with two shares, the value of the numerator will be 4. Then, we can use the same method to calculate the value of denominator $C_3(0,-2,2)$ which is 19, and so the probability of share being value -1 is $\frac{4}{19}$. Using the same method, d^0_{-2}= $\frac{3}{19}$, d^0_0=$\frac{5}{19}$, d^0_1=$\frac{4}{19}$ and d^0_2=$\frac{3}{19}$. When splitting the value $v=1$, the following is the share distributions: d^1_{-2}=$\frac{2}{18}$, d^1_{-1}=$\frac{3}{18}$, d^1_0=$\frac{4}{18}$, d^1_1=$\frac{5}{18}$ and d^1_2=$\frac{4}{18}$.

Using these distributions, we now analyse the degree of k-similarity between these two distributions. After computing all of these values, the minimum such k value is 2.375, and thus for these parameters this construction is 2.375-similar.

If $N = (k+1)M$ and $s = 3$, then the scheme satisfies k-similarity with an amplification factor of $6(k+1)$. Due to page constraints we omit the proof of these claims. We analyse the situation for $s > 3$ and the case where $t > 1$ in experimental section.

- **Computation Overhead.** One concern with this splitting scheme is that its time complexity may not be suitable for a sensor node. Computing the C values is potentially an expensive step. Dynamic programming can be used to compute the C values. We omit the details of the algorithm, but it has complexity $O(s^2N^2)$, and it is within the sensor's computation capability. That is, the values of s and N are likely to be small enough in practice to make this practical for a sensor node. A storage-computation tradeoff is possible; that is the sensor's can store the various C values.

- **Communication Overhead.** We compare our scheme with both homomorphic encryption(Castelluccia et al., 2005) and secure aggregation protocol(Frikken and Dougherty, 2008) that proposed solutions for solely privacy and integrity respectively. Specifically in our scheme, the congestion occurred in a single node is $O(s)$. The scheme in (Castelluccia et al., 2005) results in $O(1)$ congestion per node, but only provides privacy. The protocol in (Frikken and Dougherty, 2008) has $O(\triangle log N)$ congestion per node, but only provides integrity (Here, $\triangle$ is the maximum degree of aggregation tree).

7 EXPERIMENTS

In this section we describe various experiments that test the resiliency of the proposed splitting schemes. We initially focus on the case when a single share is compromised (i.e., $t = 1$), but then we consider the case when $t > 1$.

7.1 Resiliency against the Single Share Compromise

Given parameters M, s, and N the level of k-similarity can be computed as follows: Since the scheme is symmetric we need only consider the distribution of the first share. This distribution is computed as in Algorithm 1. Given this distribution it is straightforward to compute the k value by finding the i value in $[-N,N]$ and values m_0 and m_1 in $[0,M]$ that minimize:

$$\frac{\min\{D_{m_0}^{S,M,s}[i], D_{m_1}^{S,M,s}[i]\}}{\max\{D_{m_0}^{S,M,s}[i], D_{m_1}^{S,M,s}[i]\} - \min\{D_{m_0}^{S,M,s}[i], D_{m_1}^{S,M,s}[i]\}}$$

for all values $i \in [-N,N]$. This search can be expedited because this value will be minimized when $m_0 = 0$ and $m_1 = M$ (the proof of this claim is omitted due to page constraints).

We now describe the specific experiments:

- **Relation between Share Ranges and K Value.** In this part, we fix $M = 1$, $s = 3$ and varied N from 1 to 20. Figure 2 shows the k value for each value of N. First it is worth noting that this experiment validates the claims of section 6. That is if $N = (k+1)M$ that the scheme is k-resilient.

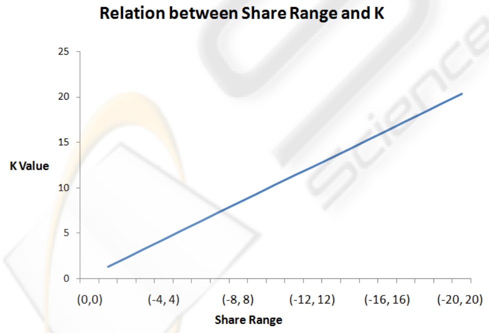

Figure 2: Relation between share range and K value.

- **Relation between the Number of Shares and K Value.** The goal of this experiment is to determine the effect of increasing the number of shares. We fixed $M = 1$, varied s from 4 to 7, and determined the minimum N value that was necessary to achieve $k = 10$. Table 1 shows the results. It appears as though the amplification factor is about the same for all values (except $s = 4$). Since increasing the number of shares increases the communication, it appears that if the adversary compromises only a single cluster head, then s should probably be chosen as 3.

Table 1: **Amplification factor for varied shares.**

s	**Minimum N for $k = 10$**	Amplification **Factor**
3	10	30.5
4	10	40.5
5	6	30.5
6	5	30.5
7	4	28.5

7.2 Resiliency against the Collusion Attack

We now consider the case where the adversary has more than one share. We generalize the definition of k-similarity, by looking at the difference in the distributions of t-shares. First, the construction actually satisfies a stronger definition of symmetry than the one presented in Definition 2 in that the distribution of any t shares is the same regardless of which shares are chosen. Thus we need only consider the distribution of the first t shares. To formalize this notion if $t = 2$, then we let $D_{m_0}^{S,M,s}[i,j]$ be the probability that two shares will be i and j when splitting the value m_0. Thus a scheme is k-similar if for all possible share values i and j and values m_0 and m_1:

$$\frac{\min\{D_{m_0}^{S,M,s}[i,j], D_{m_1}^{S,M,s}[i,j]\}}{\max\{D_{m_0}^{S,M,s}[i,j], D_{m_1}^{S,M,s}[i,j]\} - \min\{D_{m_0}^{S,M,s}[i,j], D_{m_1}^{S,M,s}[i,j]\}} \geq k$$

We start with the case where $t = 2$. Given parameters M, s, and N the level of k-similarity can be computed as follows: since the scheme is symmetric we only need to consider the pair distributions of the first and the second share, namely $Pr[s_1 = i \wedge s_2 = j](i, j \in [-N,N])$. The distribution is computed as following:

$$Pr[s_1 = i \wedge s_2 = j] = Pr[s_1 = i | s_2 = j] Pr[s_2 = j]$$

Based on Algorithm 1, we have

$$Pr[s_1 = i | s_2 = j] = \frac{C_{s-2}(v - i - j, -N, N)}{C_{s-1}(v - j, -N, N)}$$

$$Pr[s_2 = j] = \frac{C_{s-1}(v - j, -N, N)}{C_s(v, -N, N)}$$

Therefore,

$$Pr[s_1 = i \wedge s_2 = j] = \frac{C_{s-2}(v - i - j, -N, N)}{C_s(v, -N, N)}$$

Given this distribution it is straightforward to compute the k value by finding the i and j values in $[-N,N]$ and values m_0 and m_1 that minimizes the formula in Definition 10. Similar to Section 7.1, these value will be minimized when $m_0 = 0$ and $m_1 = M$.

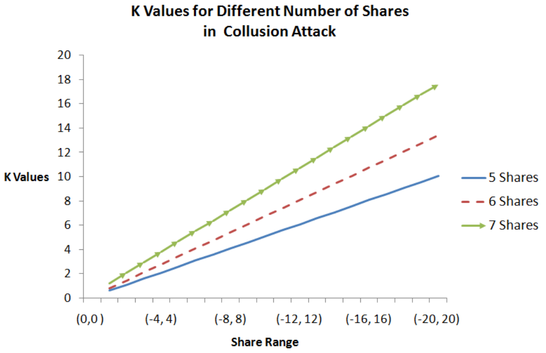

Figure 3: K values for different number of shares in collusion attack.

- **Relation between Number of Shares and K.** In this part, we fix $M = 1$, $t = 2$, varied N from 1 to 20, and varied s from 5 to 7. We did not consider s as 3 or 4, because these provide only 0-similarity. To be more specific if $v = 0$ and $s = 3$, then it is possible using the construction that 2 shares are $-N$ and 0 (the 3rd share would be N), but it is not possible to have 2 shares be $-N$ and 0 when $v = 1$ (the 3rd share would need to be $N+1$, which is not in the value share range).

  Figure 3 shows the results of this experiment. First, observe that it is possible to obtain reasonable values of k for these values. Furthermore, it appears that there is a linear relationship between N and the k value. Specifically, when $s = 5$, the scheme appears to be $.5N$-resilient, when $s = 6$, the scheme appears to be $.67N$-resilient, and when $s = 7$, the scheme appears to be $.88N$-resilient.

- **Relation between Amplification Factor and K.** In this part, we fixed $s = 5$, $t = 2$, varied M from 6 to 10, and varied N from 10 to 200. Figure 4 shows the k value for each pair of N and M values. Again the linear relationship seems to hold, and it appears as though k is linear in the value $\frac{N}{M}$. Similar results hold when s is increased (but these experiments are omitted due to page constraints).

7.2.1 The Case when $t > 2$

Due to the page limit, we omit from the manuscript the experiments for when $t > 2$ shares are compromised, but the basic idea is that, we will compute the distributions for the first t shares using the formula $Pr[s_1 = v_1 \wedge s_2 = v_2 \wedge \cdots \wedge s_n = v_n] = \frac{C_{s-n}(v-\Sigma_{i=1}^{n} v_i, -N, N)}{C_s(v,-N,N)}$, and compute the k value based on Definition 10. Similar results hold in this case, but the number of shares must be increased. That is, it must be that $s > 2t+1$ in order to have k-similarity for $k > 0$.

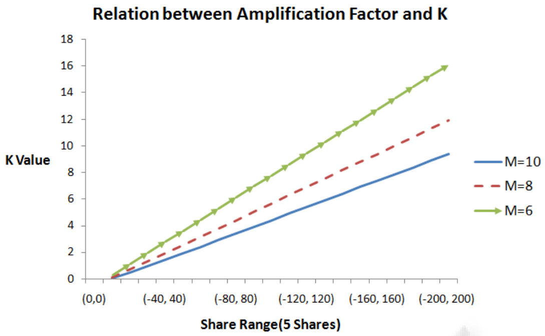

Figure 4: Relation between amplification factor and K value in collusion attack.

8 SUMMARY AND FUTURE WORK

In this paper, we introduced a new integrity measure and a new privacy measure, called called k-similarity, which is a weaker notion than cryptographic privacy, but it is still useful in real applications. Furthermore, we built a splitting construction that can achieve both the meaningful privacy and integrity during the SUM aggregation. And finally, we implemented a series of experiments to test the effectiveness of our technique against the adversaries who captured a certain number of shares. There are several problems for future work, including:

1. The splitting scheme currently only protects against leaf nodes reporting false values. While the scheme could be combined with the approach in (Roberto et al., 2009) to protect against malicious aggregator nodes, it would be desirable to create one mechanism that handles both type of integrity violations. For example, is it possible to combine the splitting scheme with the scheme in (Chan et al., 2006) by using different aggregation trees to obtain similar results.
2. The current scheme doesn't use expensive cryptography (homomorphic encryption, zero knowledge proofs, etc). Is it possible to obtain cryptographic privacy and constant amplification factor? Or, is there a different approach that does not use expensive cryptography that achieves this result?

3. The analysis in this paper was for the case when $t = 1$. However, there appears to be a linear relationship between the k and N/M. Can this result be formalized and be proven for $t > 1$?

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their comments and useful suggestions. Portions of this work were supported by Grant CNS-0915843 from the National Science Foundation.

REFERENCES

Alzaid, H., Foo, E., and Nieto, J. G. (2008). Secure data aggregation in wireless sensor network: a survey. In *AISC '08: Proceedings of the sixth Australasian conference on Information security*, pages 93–105, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Camtepe, S. A. and Yener, B. (2005). Key distribution mechanisms for wireless sensor networks: a survey. Technical report, Rensselaer Polytechnic Institute.

Castelluccia, C., Mykletun, E., and Tsudik, G. (2005). Efficient aggregation of encrypted data in wireless sensor networks. In *MOBIQUITOUS '05: Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 109–117, Washington, DC, USA. IEEE Computer Society.

Castelluccia, C. and Soriente, C. (2008). Abba: A balls and bins approach to secure aggregation in wsns. In *WiOpt'08:Sixth International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*.

Chan, H., Perrig, A., and Song, D. (2006). Secure hierarchical in-network aggregation in sensor networks. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 278–287, New York, NY, USA. ACM.

Frikken, K. B. and Dougherty, IV, J. A. (2008). An efficient integrity-preserving scheme for hierarchical sensor aggregation. In *WiSec '08: Proceedings of the first ACM conference on Wireless network security*, pages 68–76, New York, NY, USA. ACM.

He, W., Liu, X., Nguyen, H., Nahrstedt, K., and Abdelzaher, T. (2007). Pda: Privacy-preserving data aggregation in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2045–2053.

Hu, L. and Evans, D. (2003). Secure aggregation for wireless networks. In *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, page 384, Washington, DC, USA. IEEE Computer Society.

Intanagonwiwat, C., Estrin, D., Govindan, R., and Heidemann, J. (2002). Impact of network density on data aggregation in wireless sensor networks. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 457, Washington, DC, USA. IEEE Computer Society.

Kamat, P., Zhang, Y., Trappe, W., and Ozturk, C. (2005). Enhancing source-location privacy in sensor network routing. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 599–608, Washington, DC, USA. IEEE Computer Society.

Madden, S., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2002). Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146.

Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., and Culler, D. E. (2002). Spins: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534.

Przydatek, B., Song, D., and Perrig, A. (2003). Sia: secure information aggregation in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 255–265, New York, NY, USA. ACM.

Roberto, D. P., Pietro, M., and Refik, M. (2009). Confidentiality and integrity for data aggregation in wsn using peer monitoring. In *Security and Communication Networks*, pages 181–194.

Yang, Y., Shao, M., Zhu, S., Urgaonkar, B., and Cao, G. (2008). Towards event source unobservability with minimum network traffic in sensor networks. In *WiSec '08: Proceedings of the first ACM conference on Wireless network security*, pages 77–88, New York, NY, USA. ACM.

Yang, Y., Wang, X., Zhu, S., and Cao, G. (2006). Sdap: a secure hop-by-hop data aggregation protocol for sensor networks. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 356–367, New York, NY, USA. ACM.