Data Security in Unattended Wireless Sensor Networks

Roberto Di Pietro, Luigi V. Mancini, Claudio Soriente, Angelo Spognardi, and Gene Tsudik

Abstract—In recent years, Wireless Sensor Networks (WSNs) have been a very popular research topic, offering a treasure trove of systems, networking, hardware, security, and application-related problems. Much of prior research assumes that the WSN is supervised by a constantly present sink and sensors can quickly offload collected data. In this paper, we focus on *Unattended* WSNs (UWSNs) characterized by intermittent sink presence and operation in hostile settings. Potentially lengthy intervals of sink absence offer greatly increased opportunities for attacks resulting in erasure, modification, or disclosure of sensor-collected data. This paper presents an in-depth investigation of security problems unique to UWSNs (including a new adversarial model) and proposes some simple and effective countermeasures for a certain class of attacks.

Index Terms—Wireless sensor networks, data survival, mobile adversary.

1 Introduction

Ever since initially appearing on the research horizon in mid 1990s, sensors and sensor networks have received a great deal of attention from diverse CS communities, including hardware, networking, operating systems, database, security, and various application-specific areas. Wireless Sensor Networks (WSNs)—composed of a large number of small resource-limited sensors—are of particular interest. According to the large body of accumulated research literature, WSNs have many real, anticipated, and imagined applications.

Many (or even most) WSNs are assumed to operate in real-time mode wherein, soon after acquiring data, sensors communicate it to a trusted online entity, i.e., a sink. There are also other application settings where the real-time mode is not viable, due to the intermittent or sporadic sink presence. We identify some reasons for this, listed in the order of perceived likelihood:

1. As a centralized and trusted collection point, the sink represents a critical resource. It is a single point of failure and a very attractive attack target. Destroying or incapacitating the sink essentially "kills" the entire network, whereas compromising the sink yields a collection of potentially valuable data.
2. The operating environment can preclude sink's constant presence. In addition to collecting data from its constituent sensors, the sink often serves as the WSN's gateway to the outside. However, if a WSN operates in a location which is too remote, communication between the sink (if it were onsite) and the rest of the world might be impossible.
3. On a related note, if the scale (in terms of number of sensors) and/or the coverage (in terms of geographical area) of the WSN is very large, the sink needs to be itinerant.
4. As an entity involved in massive data processing and communication with a multitude of sensors, the sink requires a lot of energy. Thus, although it may be physically present at all times, the sink might need to be switched off periodically in order to conserve energy.

The entire class of WSNs with intermittent sink presence is referred to as *Unattended Wireless Sensor Networks* or UWSNs [8]. Envisaged UWSN settings include:

- A UWSN situated in a remote area of a national park monitoring firearm discharge, illicit crop cultivation, and other illegal activities.
- A UWSN deployed along an international border to monitor drug/weapons smuggling and human trafficking.
- A treaty compliance UWSN operating under a United Nations mandate in a rogue nation, in order to monitor nuclear emissions.
- A military UWSN in a battlefield setting monitoring troop movements and other enemy activity.

It is not coincidental that these examples have a common feature of deployment in hostile environments, albeit, the level of "hostility" clearly varies. Regardless of the specifics, a hostile environment implies the existence of an adversary.

- *R. Di Pietro is with the UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Spain and the Dipartimento di Matematica, Università di Roma Tre, Largo San Leonardo Murialdo, 1, 00146 Roma, Italy. E-mail: dipietro@mat.uniroma3.it.*
- *L.V. Mancini is with the Dipartimento di Informatica, Università degli studi di Roma "La Sapienza," via Salaria, 113, 00198 Roma, Italy. E-mail: lv.mancini@di.uniroma1.it.*
- *C. Soriente is with the Computer Science Department, Secure Computing and Networking Center, 458 Computer Science (CS) Building, University of California Irvine, Irvine, CA 92697-3435. E-mail: csorient@ics.uci.edu.*
- *A. Spognardi is with the Planète project, INRIA Rhône-Alpes, 655 avenue de l'Europe, Montbonnot, 38334 Saint Ismier Cedex, France. E-mail: angelo.spognardi@inrialpes.fr.*
- *G. Tsudik is with the Computer Science Department, University of California Irvine, Bren Hall, 3rd Floor, Irvine, CA 92697-3435.*

Manuscript received 2 Jun 2008; revised 27 Nov. 2008; accepted 23 Feb. 2009; published online 23 July 2009.
Recommended for acceptance by D.R. Avresky, H. Prokop, and D.C. Verma.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2008-06-0265.
Digital Object Identifier no. 10.1109/TC.2009.109.

0018-9340/09/$25.00 © 2009 IEEE Published by the IEEE Computer Society

In prior WSN security literature, adversary's goals typically include deletion, injection, and modification of data sent by sensors to the sink, or commands sent by the sink to the sensors. Other types of attacks might involve cloning of compromised sensors, creation of routing anomalies, and sleep deprivation (which causes battery depletion). Also, prior work usually assumes that some upper bounded number of sensors will be compromised over the entire lifetime of the network. (Individual sensor compromise is viable since tamper resistance is unrealistic for most WSN settings.) However, an online sink can detect and isolate compromised sensors, thus mitigating attack consequences. This line of defense is clearly impossible in a UWSN.

In this paper, we focus on UWSNs operating in hostile settings where the adversary's goals and abilities are tailored to the unattended nature of the network. We assume that the adversary (μADV from now on) can compromise at most a certain number (or fraction) of sensors **at any given time**. Awareness of sink's absence allows μADV to move between sets of compromised sensors, gradually undermining overall UWSN security. Assuming that μADV needs certain time to compromise one set of sensors and migrate to the next, the time between successive sink visits can be viewed as a sequence of compromise intervals or rounds. Regardless of its goals (which might vary as discussed below), our adversary's most distinctive feature is its mobility—the ability to seamlessly move between sets of compromised sensors.

A seemingly similar *mobile adversary* is well known in the theoretical cryptography community, since the pioneering work of Ostrovsky and Yung in 1991 [24]. However, as discussed in Section 2.2 below, there are some major differences between our envisaged mobile adversary and its preexisting cryptographic counterpart.

This paper makes several contributions. First, it defines and explores a new mobile adversary model unique to UWSNs. This model turns out to have multiple "flavors," depending on specific attack goals. Second, it develops and evaluates several techniques that mitigate mobile adversary attacks aimed at both targeted and indiscriminate erasure of data. Finally, it opens up new research directions and identifies challenges in the context of UWSN security.

The rest of the paper is organized as follows: Section 2 introduces our assumptions, adversarial model, and proposed defense strategies. Next, Sections 3 and 4 analyze effectiveness of our solutions in the presence of an adversary focused on erasing sensor-collected data. Section 5 shows how replication can further increase the odds of data survival. Section 6 overviews relevant prior work, and Section 7 provides conclusions and future research directions. Finally, Appendix A compares a fragmentation-based approach with simple data replication.

2 Preliminaries

This section presents our assumptions about the network and the anticipated adversary, and defines the scope of this paper. Table 1 summarizes our notation.

2.1 Network

We assume that the network is composed of a large number (e.g., hundreds or thousands) of homogeneous sensors. The network is always connected: any two sensors can communicate, either directly or via other sensors.

TABLE 1
Notation

μADV	mobile adversary
n	# of sensors, i.e., the size of the UWSN
i, j	sensor indexes
s_i	sensor i
r	round index
x	target data
$\check{s}_r$	sensor in possession of x at the start of round r
$\mathring{s}_r$	sensor that receives x during round r
v	maximum number of rounds between successive sink visits
k	maximum number of concurrently compromised sensors
C_r	set of compromised sensors at round r

The network is unattended most of the time. A global parameter v represents the maximum number of rounds between consecutive sink visits. At each visit, the sink collects all data from all sensors. Upon collection, each sensor erases all previously collected data and securely reinitializes all sensors.

Each sensor acquires one data unit per round and has enough storage to accommodate $O(v)$ sensed data. Also, each sensor has a pseudorandom number generator. No other cryptographic facilities are assumed at this point.

All sensors are assumed to have loosely synchronized clocks [12]. Sensors are programmed to acquire data from the environment at fixed intervals. (We use the terms *interval* and *round* interchangeably.) In this paper, we do not consider networks that operate on a query basis, i.e., where data acquisition is performed only upon explicit request by the sink. This is because, in such networks, sensors do not accumulate data during sink absence and only acquire data when the sink is present.

At least initially, we are not concerned with power consumption, since our primary goal is data security and survivability. However, proposed solutions do take energy consumption into account.

2.2 Adversary

We envision a powerful mobile adversary. One important feature that separates it from other adversarial models is **mobility**. We assume that μADV can compromise a subset (up to a certain size) of sensors within a particular time interval. However, we do not assume that the subset of compromised sensors is clustered or contiguous, i.e., concurrently compromised sensors can be spread through the entire network. Furthermore, in the next interval, μADV can migrate and compromise a different subset.[1] Given enough intervals, μADV can gradually subvert the entire UWSN. While it occupies a given sensor, μADV can read from, and possibly write to, the compromised sensor's storage and any of its communication interfaces. It can thus learn all the sensor's secrets as well as eavesdrop on all incident communication.

Based on our discussion thus far, μADV resembles its dual well known in the cryptographic literature as the

[1] There is no requirement for μADV to move; however, mobility is clearly in its interest.

TABLE 2
Summary of Adversarial Types

	Curious	Polluter	Search-and-Erase	Search-and-Replace	Eraser
Visible	N/A	Reactive	Reactive or Proactive	N/A	Reactive
Stealthy	Reactive	N/A	Reactive or Proactive	Reactive or Proactive	N/A

mobile adversary [24]. An entire branch of cryptographic research, called *Proactive Cryptography*, has been developing cryptographic techniques that preserve security in the presence of this mobile adversary, e.g., [11], [27]. However, there is one crucial difference: the goal of the cryptographic mobile adversary is to discover some systemwide secret (usually, a decryption or a signature key) which has been predistributed—via secret-sharing techniques [28]—among the system components. Whereas, our mobile adversary's goal is to read, erase, or modify data collected by unattended sensors. (There is, indeed, no systemwide secret in our context.) Consequently, research results in proactive cryptography do not apply to the problem at hand.

We consider several μADV flavors, each with different goals.

Curious. Aims to learn as much sensed data as possible. In general, it is not difficult to read data from both RAM and ROM of a commodity sensor, as demonstrated in [6]. If no countermeasures are taken, μADV can simply compromise sensors and learn the data directly. Of course, μADV might be focused on learning specific sensor measurements that represent critical or high-value data.

Polluter. Aims to mislead or confuse the sink by introducing fraudulent data into the network. Such data may change sensing statistics and, as a consequence, affect sink's (and higher level) actions. Note that a polluter does not alter any existing measurements.

Search-and-Erase. Aims to prevent certain target data from reaching the sink. Consider, for example, a sensor network monitoring nuclear emissions where the sink raises an alarm if one of the sensors reports a value above a certain threshold. μADV's goal is to find that value and erase it before it reaches the sink. If we assume that the sink tolerates some missing measurements (due to occasional errors or malfunctions), μADV will remain undetected even if it succeeds in erasing the target data.

Search-and-Replace. If the sink has no tolerance for lost data (and μADV knows this), the corresponding model changes from Search-and-Erase to Search-and-Replace. This μADV also aims to prevent some target data from reaching the sink; however, it wants to replace the target data with some concocted value.

Eraser. Aims to indiscriminately erase as much data as possible, i.e., its main goal is denial-of-service.

It is easy to envision other types of adversarial behavior in UWSNs. Any combination of the above types is possible and viable. In practice, μADV does not have to neatly fit into the pigeon holes outlined above. However, we focus on these basic types of adversarial behavior, since successful mitigation of their respective attacks will allow us to combine techniques and address most, or even all, hybrid variants.

We also acknowledge that nothing prevents μADV from physically destroying or damaging sensors, especially, since the network is unattended most of the time. However, such crude behavior leaves physical evidence and we assume it to be in μADV's interest to be subtle and, whenever possible, stealthy. This means that, depending on the type of attack, μADV strives to remain undetected. If it succeeds in doing so, its movements become not only unpredictable but also untraceable. In particular, it might be impossible to detect if and when μADV ever compromised a particular sensor.

As evident from the description of μADV types, stealth is not always possible. In other words, the ability to remain undetected is based on the μADV's goal. Clearly, a Curious μADV is expected to be invisible. A Search-and-Erase μADV might take advantage of knowing that the sink has a certain tolerance for missing data and also remain stealthy. The Search-and-Replace μADV is even more likely to remain stealthy since, if it succeeds, the sink will detect no missing (and no extra) data. Whereas, neither Polluter nor Eraser can avoid detection, since their goal is pure denial-of-service.

We further distinguish between a *proactive* and a *reactive* adversary. The latter is assumed to be dormant (inactive) until it gets a signal to respond to certain target data. As soon as this happens, μADV *reacts* and starts compromising sensors in order to accomplish its goal. In contrast, a *proactive* μADV roams the network ahead of time, compromises subsets of sensors and waits for a signal to respond to certain target data. A proactive μADV, as we discuss below, has some definite advantages over its reactive counterpart.

Table 2 summarizes the adversary's potential based on attack goals and visibility.

2.3 Scope and Defense Strategies

Although all adversarial types outlined above have some features in common, their unique goals call for distinct defense strategies.

It is well known that various cryptographic techniques (e.g., signatures, encryption, MACs) are quite effective against certain attacks. Nevertheless, cryptography comes at a cost. While symmetric cryptography is generally considered viable for sensors, public key cryptography is considered too expensive in terms of energy and computational costs. Even symmetric cryptography can be burdensome, since it requires sophisticated (and scalable) key distribution and management techniques.

Even if cost were not an issue, the use of cryptography is not a panacea. As illustrated in other recent work [23], [7], [8], public key cryptography is only effective (i.e., offers additional security) if coupled with a per-sensor True Random Number Generator (TRNG). Also symmetric cryptography is practically useless, unless key evolution is employed to obtain forward secrecy, which offers some defense against a reactive μADV. Unfortunately, symmetric cryptography offers no protection against a proactive μADV, even with a per-sensor TRNG.[2]

In the rest of this paper, we explore the extent to which some anticipated adversarial types can be addressed **without cryptography**. We show that simple data migration and dissemination protocols can be quite effective against Search-and-Erase and Eraser. In doing so, we do not rule out or ignore cryptography altogether; we merely consider it as a complementary approach.

In terms of potential defense strategies, we identify several possibilities based on data migration and dissemination:

- DO-NOTHING (DN): The default and the easiest option is to do nothing: simply leave data resident on the sensor that collected it and wait for the sink.
- MOVE-ONCE (MO): A trivial alternative is for each sensor—right after collection—to move newly obtained data to some randomly picked sensor. Data then remain at their new *home* until the next sink visit.
- KEEP-MOVING (KM): A more laborious option is to move data continuously, i.e., at every interval, each sensor moves each data item, individually, to another randomly chosen sensor.

No matter what defense strategy is used, we assume that μADV is fully aware of it. Knowing the network's strategy, μADV can formulate an attack strategy that maximizes its chances of reaching its goals.[3] In general, μADV's strategy defines how it selects the set of sensors to compromise at any round. Sending data to random peers requires for sensors to be aware of the network topology. We need this assumption to simplify the model for the initial foray into this area. One alternative is to adopt a geographical routing approach, whereby random peers are picked based on their physical positions (coordinates) [2], [20]. Furthermore, we assume that the network is not subject to message dropping or wormhole attacks; countermeasures for these attacks can be found in [5], [13], [19].

In the remainder of the paper, we investigate the effectiveness of aforementioned defense strategies against Search-and-Erase and Eraser. We provide extensive analysis and support them with simulation results.

3 Search-and-Erase

We now consider a reactive Search-and-Erase. This adversary type learns the identity of the sensor during the round when the target value is acquired. In the next round, μADV begins compromising subsets of sensors. To give μADV greatest advantage, we assume that the target value is sensed at round 0, so that μADV has v rounds to delete it before the next sink visit. We also assume that sensors (i.e., the network as a whole) are unaware of which data μADV is pursuing; thus, all data must be protected equally.

3.1 DN

With the DN strategy, a sensor acquires a value and stores it locally, where it remains until the next sink visit. Search-and-Erase succeeds very quickly: it learns $\check{s}_0$ (the identity of the target sensor) at the end of round 0. At round 1, it compromises any set of sensors that includes $\check{s}_0$ and deletes x.

3.2 MO

With the MO strategy, at every round, each sensor offloads its newly acquired value to a randomly selected peer. The latter stores the value until the next sink visit. Even though μADV learns $\check{s}_0$ at the end of round 0, it has no knowledge of $\mathring{s}_0$—the sensor selected by $\check{s}_0$ as *home* for the target data. Since any sensor is equally likely to be $\mathring{s}_0$, μADV can do no better than select C_1 at random.

If $\check{s}_1 \in C_1$, then μADV wins at round 1. Otherwise, since target data do not migrate further, μADV's best strategy is to minimize the number of rounds needed to inspect all sensors. Thus, μADV proceeds, in each round r, to select C_r such that $C_r \cap (C_1 \cap \cdots \cap C_{r-1}) = \emptyset$. Assuming (without loss of generality) that n is divisible by k, after $\frac{n}{k}$ rounds, μADV "visits" each sensor. On average, $\frac{n}{2k}$ rounds are enough to find and delete the target data.

We now consider the probability of μADV winning at some round $r \leq \frac{n}{k}$. We express the probability of event G_r = "μADV finds target data at round r" at round $1 \leq r \leq \frac{n}{k}$, conditioned upon the event F_{r-1} = "target data are not found in prior $r-1$ rounds," as

$$Pr[G_r|F_{r-1}] = \frac{k}{n-(r-1)k}. \qquad (1)$$

To clarify the phenomenon in question, we also consider the probability $Pr[G_r]$ which can be expressed as

$$\begin{aligned} Pr[G_r] &= Pr[G_r|F_{r-1}]Pr[F_{r-1}] + Pr[G_r|\overline{F_{r-1}}\ ]Pr[\overline{F_{r-1}}] = \\ &= Pr[G_r|F_{r-1}]Pr[F_{r-1}] + 0 = \\ &= Pr[G_r|F_{r-1}]Pr[F_{r-1}|F_{r-2}]Pr[F_{r-1}] = \\ &= Pr[G_r|F_{r-1}]Pr[F_{r-1}|F_{r-2}]Pr[F_{r-1}|F_{r-2}]Pr[F_{r-2}]\ldots \\ &\quad \ldots Pr[F_1] = \\ &= \left[\prod_{i=1}^{r-1}\left(1-\frac{k}{n-(i-1)k}\right)\right]\frac{k}{n-(r-1)k} = \\ &= \left(1-\frac{k}{n}\right)\left(1-\frac{k}{n-k}\right)\left(1-\frac{k}{n-2k}\right)\cdots \\ &\quad \cdots\left(1-\frac{k}{n-(r-2)k}\right)\frac{k}{n-(r-1)k} = \\ &= \left(\frac{n-k}{n}\right)\left(\frac{n-k-k}{n-k}\right)\left(\frac{n-2k-k}{n-2k}\right)\cdots \\ &\quad \left(\frac{n-(r-2)k-k}{n-(r-2)k}\right)\frac{k}{n-(r-1)k} = \frac{k}{n}. \end{aligned}$$

The above equation provides a curious result: at every round, μADV's probability of finding and deleting x is $\frac{k}{n}$. It might appear counterintuitive as it is natural to expect it to increase as rounds go by.

[2] In fact, as shown in [23], [7], the only "salvation" in case of symmetric cryptography is through some form of sensor cooperation to attempt recovery from past compromise.

[3] The reverse is not true, i.e., we *do not* assume that the adversarial strategy is known to the UWSN.

3.3 KM

We now investigate the KM strategy, whereby, at the end of every round, each sensor moves all of its data to randomly selected sensors. Algorithm 1 shows the corresponding pseudo-code run by μADV.

Algorithm 1. KM

```
      /* start round 0 */
      all sensors acquire their values
      each sensor exchanges data with others
0     $\mu$ADV learns $\check{s}_0$ and $x$
      /* end round 0 */
      SET r=1
      SET found=FALSE
      while (($r \leq v$) and (not found)) do
          /* start round r */
1         select $C_r$ /* new set of sensors to compromise */
2         release $C_{r-1}$ and compromise $C_r$
3         if ($\check{s}_r \in C_r$) then
3.1           delete $x$
3.2           SET found=TRUE
          all sensors sense their values
          each sensor exchanges data with others peers
4         if ($\mathring{s}_r \in C_r$) then
4.1           delete $x$
4.2           SET found=TRUE
          $r = r + 1$
          /* end round r */
```

As shown in Algorithm 1, $\check{s}_0$ is the sensor that acquired target data. In any round $r \geq 0, \check{s}_r$ denotes the sensor in possession of x until step 4 and $\mathring{s}_r$—the sensor that receives and stores x in step 4. $\mathring{s}_r$ keeps x until step 4 of the following round.

Clearly, it is in μADV's interest to keep moving and compromise a different set of sensors at each round. That is, μADV chooses C_r such that $C_r \cap C_{r-1} = \emptyset$. Note that, at each round, μADV has two chances to find x: 1) when it compromises a new set of sensors, since one of them might store x, and 2) at the end of the round, when messages are exchanged, if one of the currently compromised sensors receives x. Thus, if $C_r \cap C_{r-1} \neq \emptyset, \mu$ADV misses chance 1) for every sensor in the intersection of the two sets.

We refer to the μADV who chooses C_r such that $C_r \cap C_{r-1} = \emptyset$, as Frantic.

Theorem 3.1. *The probability that x survives v rounds in the presence of a Frantic μADV is*

$$P_f(v) = P_1 \cdot P_2^{v-1} \cdot P_3^{v-1}, \tag{2}$$

where

$$P_1 = \left(1 - \frac{k}{n}\right)^2, \quad P_2 = \left(1 - \frac{k}{n}\right), \quad \text{and} \quad P_3 = \left(1 - \frac{k}{n-k}\right).$$

Proof. Let X_v be the random variable that assumes value 1 if x survives at round v, and 0 otherwise. We then have

$$\begin{aligned} Pr[X_v = 1] &= Pr[X_v = 1 \wedge (X_{v-1} = 1 \vee X_{v-1} = 0)] = \\ &= Pr[X_v = 1 \wedge X_{v-1} = 1] + Pr[X_v = 1 \wedge X_{v-1} = 0] = \\ &= Pr[X_v = 1 \wedge X_{v-1} = 1] + 0 = \\ &= Pr[X_v = 1 | X_{v-1} = 1] \cdot Pr[X_{v-1} = 1]. \end{aligned}$$

In other words, x survives in round v if and only if μADV fails to capture x in all prior rounds. Similarly, for all rounds, we have

$$\begin{aligned} Pr[X_v = 1] = Pr[X_v &= 1 | X_{v-1} = 1] \cdot \\ &\cdot Pr[X_{v-1} = 1 | X_{v-2} = 1] \cdot \ldots \\ &\ldots \cdot Pr[X_2 = 1 | X_1 = 1] \cdot Pr[X_1 = 1] = \\ &= Pr[X_2 = 1 | X_1 = 1]^{v-1} \cdot Pr[X_1 = 1]. \end{aligned} \tag{3}$$

The last relation holds since the respective probabilities for each round are the same and independent from v. We further have

$$Pr[X_2 = 1 | X_1 = 1] = Pr[\{\check{s}_2 \notin C_2\}] \cdot Pr\left[\{\mathring{s}_2 \notin C_2\}\right]. \tag{4}$$

Given that x survived in the first round and considering that $C_1 \cap C_2 = \emptyset$, it follows that

$$Pr[\{\check{s}_2 \notin C_2\}] = 1 - \frac{k}{n-k} = P_3. \tag{5}$$

Similarly, the probability that no sensor in C_2 receives x is

$$Pr[\{\mathring{s}_2 \notin C_2\}] = 1 - \frac{k}{n} = P_2. \tag{6}$$

Substituting (5) and (6) in (4) we obtain:

$$Pr[X_2 = 1 | X_1 = 1] = \left(1 - \frac{k}{n}\right)\left(1 - \frac{k}{n-k}\right) = P_2 \cdot P_3.$$

Our claim follows, based on Equation (2):

$$P_f(v) = Pr[X_v = 1] = P_1 \cdot P_2^{v-1} \cdot P_3^{v-1}.$$

□

We observe that, to achieve $C_r \cap C_{r-1} = \emptyset, \mu$ADV can randomly select two disjoint sets C_1 and C_2 and alternate between them. In other words, there is no need for μADV to move around the entire network. Since the probability of any sensor being chosen as a recipient is uniform, this type of μADV—which we refer to as Smart—has exactly the same probability of finding x as the Frantic μADV. We also note that, since the UWSN might be deployed over a wide geographical area, compromising some sensors might require more effort than others. Therefore, the Smart μADV can restrict its activity to the most accessible (easiest to compromise) sensors $C_1 \cup C_2$ and thus minimize physical mobility.

Fig. 1 shows that survival probability at each round is the same for both Frantic and Smart μADV-s.

3.4 MO versus KM: Expected Winning Round

We now show that μADV's expected winning round is the same for both MO and KM defense strategies. We model the system by a Markov Chain with the following states:

- S_0: Represents the network at round 0, when μADV has not yet chosen any sensors to compromise. This is also the only initial state of the chain.
- S_r: Represents the network at round $r > 0$, when μADV corrupted a set of sensors C_r and is still looking for the target value, meaning that x was

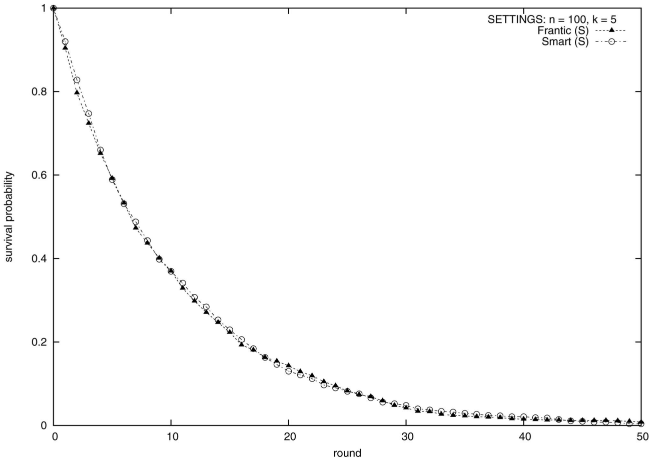

Fig. 1. Comparison of the Frantic and the Smart adversary; network strategy is KM. (S) stands for simulated results.

stored elsewhere and, after message exchange, no sensor in C_r received it.

- S_c: Represents the network at round $r > 0$, when μADV corrupted C_r and has found the target value.

The system starts in state S_0 and leaves it at round 1. It then reaches state S_r if $\check{s}_1 \notin C_1$ and $\mathring{s}_1 \notin C_1$; otherwise, it moves to S_c and the game ends.

At any subsequent round, the chain will move to:

- S_c, if μADV finds the target value;
- S_r, if μADV does not compromise the sensor storing the target value and no currently compromised sensor receives it.

The number of rounds the target value survives is the number of transactions for the chain to reach S_c.

From the analysis in Section 3.3, we build the transition matrix M of the Markov chain for the KM strategy, where each element m_{ij} represents the probability that the chain moves from state S_i to state S_j:

$$M = \begin{pmatrix} m_{00} & m_{0r} & m_{0c} \\ m_{r0} & m_{rr} & m_{rc} \\ m_{c0} & m_{cr} & m_{cc} \end{pmatrix} = \begin{pmatrix} 0 & P_1 & 1 - P_1 \\ 0 & P_2 \cdot P_3 & 1 - P_2 \cdot P_3 \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that S_c is an absorbing state ($m_{cc} = 1$) and that S_0 is a transient state, since the system cannot return to it ($m_{i0} = 0$, $\forall i$). Since the Markov chain is absorbing, the average absorbing time represents the expected μADV's winning round.

To easily compute the average absorbing time of the chain [15], we evaluate its *fundamental matrix* where each entry gives the expected number of times that the process is in a nonabsorbing state. Based on M, the fundamental matrix is

TABLE 3
μADV's Expected Winning Round

$n = 100$	$k = 2$	$k = 5$	$k = 10$
MO	25	10	5
KM	25.01	10.025	5.05

$$N = \begin{pmatrix} 1 & -P_1 \\ 0 & 1 - P_2 \cdot P_3 \end{pmatrix}.$$

Combining the fundamental matrix with the fact that the chain always starts from state S_0, we obtain the μADV's expected winning round with the KM strategy:

$$Exp = 1 + \frac{P_1}{1 - P_2 \cdot P_3}. \qquad (7)$$

For the MO strategy, it is trivial to see that the expected winning round is $\frac{n}{2k}$.

Table 3 shows μADV's expected winning round according to the previous analysis. It is clear that, on the average, both strategy provide the same survival rate of the target data.

3.5 Communication and Storage Costs

As shown in Table 4, MO and KM strategies introduce both storage and communication costs. As expected, communication overhead is higher for the latter. As far as storage overhead (determined by the number of messages a sensor receives in a single round), Table 4 shows the probability bounds that limit the queue sizes. The bounds show that proposed defense strategies are viable.

As for the maximum number of messages that could be sent to a single sensor, since each sensor selects its recipient peer uniformly and at random, this problem has a natural correspondence with the balls-and-bins model. We leverage the fact that when throwing n balls uniformly at random over n bins, the maximum load of a bin is $O(\log n)$ with probability at most $1/n$ [16]. That is, for the MO strategy, the maximum number of messages sent to a single sensor in one round is $O(\log n)$.

We also provide a bound for the probability that the number of data items stored at a sensor do not exceed a certain value ℓ. Let L_i^r be a random variable representing the number of data items stored by sensor s_i at round r. Clearly, $E[L_i^r] = r$. Applying the method of bounded differences (a special case of Azuma's inequality) [16], we have that, for $\ell > r + \sqrt{rn}$:

$$Pr\big[L_1^r \geq \ell \cup \cdots \cup L_n^r \geq \ell\big] \leq nPr\big[L_1^r \geq \ell\big] \leq e^{-r/2 + \ln n}.$$

TABLE 4
Overhead Comparison

strategy	*msg per round (r)*	*msg tot*	*stored data*	*received messages*
MO	n	$v \cdot n$	$Pr[\exists\ s_i \mid L_i^r \geq r + \sqrt{nr}] \leq e^{-r/2+\ln n}$	$O(\ln n)$
KM	vn	$(r^2/2)n$	$Pr[\exists\ s_i \mid L_i^r \geq 2er] \leq 2^{-r+\ln n}$	$Pr[\exists\ s_i \mid M_i^r \geq 2er] \leq 2^{-r+\ln n}$

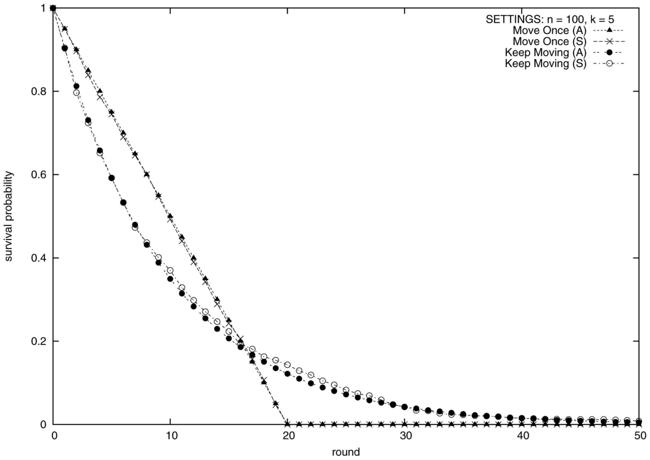

Fig. 2. Survival rate for different defense strategies against Search-and-Erase. (S) and (A) stand for simulated and analytical results, respectively.

As for the KM strategy, using the same notation, we still have that

$$Pr\big[L_1^r \geq \ell \cup \cdots \cup L_n^r \geq \ell\big] \leq nPr\big[L_1^r \geq \ell\big].$$

However, with KM, the random variables: $L_1^1, \ldots, L_1^r$ are independent; hence, we can apply the Chernoff bound [16], obtaining $(E[L_1^r] = r)$: $Pr[L_1^r \geq \ell] \leq 2^{-r}$ for $\ell > 2er$. Therefore:

$$Pr\big[L_1^r \geq \ell \cup \cdots \cup L_n^r \geq \ell\big] \leq nPr\big[L_1^r \geq \ell\big] \leq 2^{-r+\log_2 n}.$$

If we define the random variable M_i^r (which represents the number of messages received by s_i at round r), then with the same technique used above, we have

$$Pr\big[M_1^r \geq \ell \cup \cdots \cup M_n^r \geq \ell\big] \leq nPr\big[M_1^r \geq \ell\big] \leq 2^{-r+\log_2 n}.$$

3.6 Summary

The above analysis makes it clear that the choice between MO and KM depends on the frequency of sink visits. Fig. 2 shows target data survival probability against Search-and-Erase, given MO and KM defense strategies. With the former, μADV is guaranteed to win in at most $\frac{n}{k}$ rounds, whereas, with KM, target data survives somewhat longer.

Nevertheless, for the first $\frac{n}{k}$, MO performs better than KM. This is because the latter changes the location of target data at each round and, as discussed earlier, it affords μADV two chances to capture x in each round. Moreover, KM is obviously more expensive than the MO. We conclude that, if $v < \frac{n}{k}$, MO is the most effective and efficient strategy against Search-and-Erase.

4 Eraser

Recall that Eraser is not focused on any particular data. Its main goal is to delete as much data as possible before the next sink visit. As in Section 3, we assume that μADV starts compromising sensors at round 1 and has v rounds at its disposal. We investigate the effectiveness of data dissemination strategies of Section 2.3 and estimate the amount of data surviving in network at each round.

4.1 DN

To maximize its advantage, μADV needs to always compromise the set of k sensors with the highest number of stored data items. It is easy to see that this is best achieved by moving in a round-robin fashion for the first $\frac{n}{k}$ rounds. In doing so, μADV deletes $(r+1) \cdot k$ messages during the first $r \leq \frac{n}{k}$ rounds. At round $\frac{n}{k}+1$, the k sensors that have the largest number of data items are those that have not been visited for the longest time, i.e., those compromised at round 1. Each of those sensors has by now accumulated $\frac{n}{k}$ values so that μADV can delete n values, total. Using the same argument for subsequent rounds, μADV needs to move in a round-robin fashion at any round $r > \frac{n}{k}$ and to delete n values per round. As n new measurements are introduced at every round, after round $\frac{n}{k}$ the number of values in the network remains stable. Thus, the number of remaining data items at round r is

$$D_{DN}(r) = n,$$

if $r = 0$, while

$$D_{DN}(r) = n - k \cdot \left(\min\left\{r, \frac{n}{k}\right\}\right) + \sum_{i=1}^{\left(\min\left\{r, \frac{n}{k}\right\}\right)} (n - i \cdot k),$$

otherwise.

4.2 MO

Using an argument similar to that in Section 4.1, with the MO strategy, μADV needs to move in a round-robin fashion to assure that, at any round, it compromises the set of sensors with the largest number of data items.

Let $p_{MO}(r)$ be the probability that a given data item survives r rounds:

$$p_{MO}(r) = \left(1 - \frac{k}{n}\right) \prod_{i=0}^{\min\{r-1, \frac{n}{k}-1\}} \left(1 - \frac{k}{n - ik}\right). \qquad (8)$$

The component on the left is the probability that the data were sent to a noncompromised sensor. The factor with $i = 0$ represents the probability that the data have been acquired by a noncompromised sensor. The factors with $i \geq 1$ account for the probability that μADV did not compromise the sensor holding the value, i rounds after it has been sensed. The product has at most $\frac{n}{k}$ factors because the value migrates just once and μADV is guaranteed to delete it by round $\frac{n}{k}$.

Thus, the average number of values remaining at round r is

$$D_{MO}(r) = \begin{cases} n, & \text{if } r = 0, \\ n \cdot \dfrac{p_{MO}(r)}{\left(1 - \frac{k}{n}\right)} + \displaystyle\sum_{i=1}^{r} (n \cdot p_{MO}(i)), & \text{otherwise.} \end{cases} \qquad (9)$$

4.3 KM

With the KM strategy, at each round, each sensor offloads all of its stored data to randomly chosen peers. Thus, μADV is no longer sure that the least recently visited sensors are the ones holding the largest number of data items. In this scenario, μADV does not need to move in a round-robin fashion. However, it is still in its interest to choose C_r such that $C_r \cap C_{r-1} = \emptyset$.

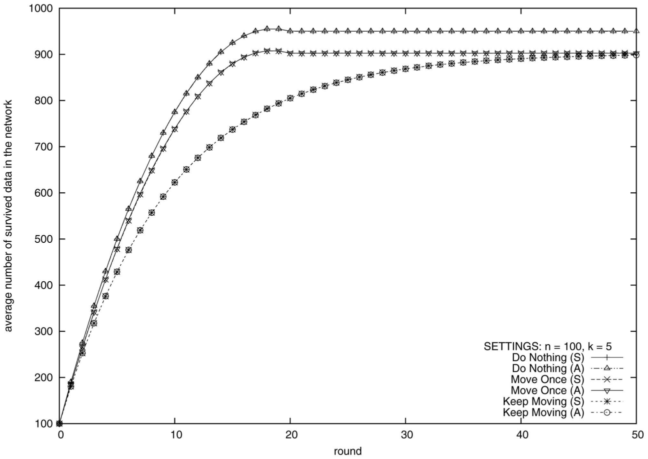

Fig. 3. Survival rate of different defense strategies against Eraser. (S) and (A) stand for simulated and analytical results, respectively.

Let $p_{KM}(r)$ be the probability that a particular value survives r rounds:

$$p_{KM}(r) = \left(1 - \frac{k}{n}\right)^2 \left(\left(1 - \frac{k}{n-k}\right) \cdot \left(1 - \frac{k}{n}\right)\right)^{r-1}. \qquad (10)$$

The squared factor is the probability that a value is sensed by, and sent to, some noncompromised sensor. The factor exponentiated to the power of $r-1$ accounts for the probability that μADV did not compromise the sensor holding that value, nor any of the comprised sensors received it, during subsequent $r-1$ rounds.

To estimate the number of data items surviving Eraser at round r, we need to take into account data sensed at any round $0 \leq i \leq r$, and the probability that they survived until the current round. The number of data items in the network at round r is

$$D_{KM}(r) = \begin{cases} n, & \text{if } r = 0, \\ n \cdot p_{KM}(r) + \sum_{i=1}^{r} (n \cdot p_{KM}(r)), & \text{otherwise.} \end{cases} \qquad (11)$$

4.4 Summary

It might seem surprising that the most effective strategy against an Eraser is DN. Fig. 3 shows that data survival is maximized if acquired data are kept *in situ*. Indeed, if data do not migrate, μADV can only erase items found in the storage of compromised sensors. If data are moved around the network, either once or at any round, μADV can delete items found in compromised sensors, as well as items that those sensors receive from their noncompromised counterparts.

In the same figure, the number of remaining data items with the DN and MO strategies, exhibit a sudden drop of k messages at round $\frac{n}{k}$. This is because μADV starts compromising sensors at round 1, while sensing starts at round 0. Nevertheless, by round $\frac{n}{k}$, μ ADV deletes all data sensed at round 0, and this *one-round advantage* is lost.

5 Replication

In this section, we investigate the effects of data replication. Along with data migration, data replication is a natural and intuitive technique for increasing the odds of data survival. If we assume that each data item is replicated R times and each replica is treated independently, the end result is an R-fold increase in storage and communication costs. In this paper, we assume all data are treated equally within the network; however, in case sensors are aware of the relevance of the sensed data, they might replicate data according to some prioritization policy. For example, in a fire-prevention monitoring sensor network, sensors might only replicate measurements that are above a critical threshold. This way, the overhead incurred in data storage and communication would be sensibly decreased. In the following, we show the gain in data survival that can be achieved with replication.

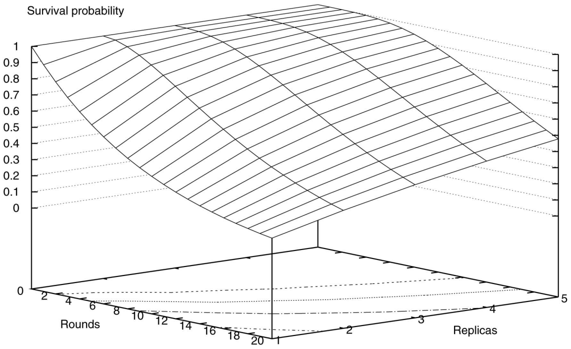

Fig. 4. Survival of Replicated Data against Search-and-Erase using the Frantic strategy; network strategy is KM.

5.1 Search-and-Erase

In order to succeed, a Search-and-Erase must delete all R copies of target data. Let $X_{i,j} = 1$ denote the event of ith replica surviving up to round j, and $X_{i,j} = 0$ denote the event of μADV erasing it by round j. The probability $\overline{P_R^v}$ of no replicas surviving up to round v is

$$\overline{P_R^v} = Pr[X_{1,v} = 0 \wedge \cdots \wedge X_{R,v} = 0] = Pr[X_{1,v} = 0]^R.$$

Then, the probability of at least one replica surviving to round v is

$$P_R^v = 1 - \overline{P_R^v}. \qquad (12)$$

With the MO strategy, using the results from Section 3.2, we have that $Pr[X_{i,j} = 1] = \frac{k}{n-jk}$. Plugging this into (12) yields

$$P_R^v = 1 - \left(1 - \prod_{i=0}^{v-1}\left(1 - \frac{k}{n-ik}\right)\right)^R. \qquad (13)$$

With the KM strategy, according to (2):

$$Pr[X_{i,j} = 1] = P_1 \cdot P_2^{j-1} \cdot P_3^{j-1}.$$

Using the probability in (12):

$$P_R^v = 1 - \overline{P_R^v} = 1 - \left(1 - P_1 \cdot P_2^{v-1} \cdot P_3^{v-1}\right)^R. \qquad (14)$$

Fig. 4 shows how data survival rate increases with replication. With no replication ($R = 1$) data survives 20 rounds with probability 0.122; at the same round, if $R = 5$, probability of at least one copy surviving is 0.48.

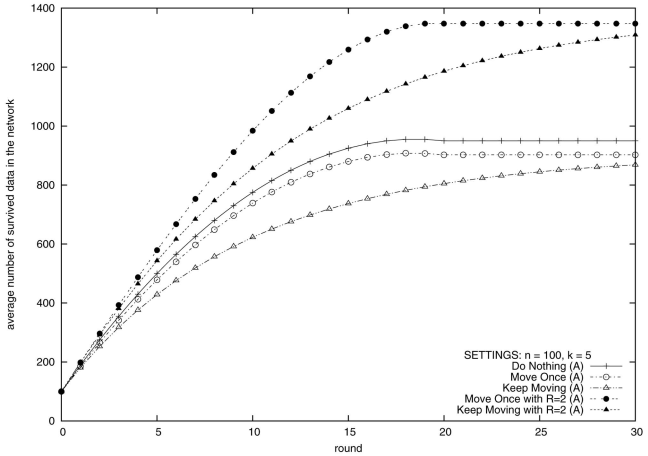

Fig. 5. Average number of data items surviving Eraser, with and without replication. (A) stands for analytical results.

5.2 Eraser

Since Eraser deletes all data it encounters, replication naturally increases survival rate for data items acquired by any sensor at any round. Effectiveness of replication is estimated by the number of distinct data items that survive up to round v (replicas of the same data are not taken into account).

5.2.1 MO Strategy

With the MO strategy, the probability of all R replicas of a given data item being erased by round r is $(1 - p_{MO}(r))^R$. Thus, the probability of at least one replica surviving at round r is $1 - (1 - p_{MO}(r))^R$.

Using an argument similar to that in Section 3.3, we can derive the overall number of distinct data items that survive by round r:

$$D^R_{MO}(r) = \begin{cases} n, & \text{if } r = 0, \\ \sum_{i=0}^{r} n \cdot (1 - (1 - p_{MO}(i))^R), & \text{otherwise.} \end{cases} \tag{15}$$

5.2.2 KM Strategy

With R-fold replication, the probability of at least one replica surviving r rounds is $1 - (1 - p_{KM}(r))^R$. Here, we also use the argument of Section 3.3 to estimate the overall number of distinct data items surviving by round r:

$$D^R_{KM}(r) = \begin{cases} n, & \text{if } r = 0, \\ n(1 - (1 - p_{KM}(r))^R) + \\ \quad + \sum_{i=1}^{r} n \cdot (1 - (1 - p_{KM}(i))^R), & \text{otherwise.} \end{cases} \tag{16}$$

Fig. 5 shows how data migration and replication together affect data survival against Eraser. As shown in Section 4, without replication, DN is the best strategy against this kind of adversary. However, data migration improves data survival even with a single replica (i.e., $R = 2$).

Fig. 6 shows how replication coupled with the KM strategy improves data survival. After 20 rounds, without replication, μADV erases 80 percent of all data, whereas,

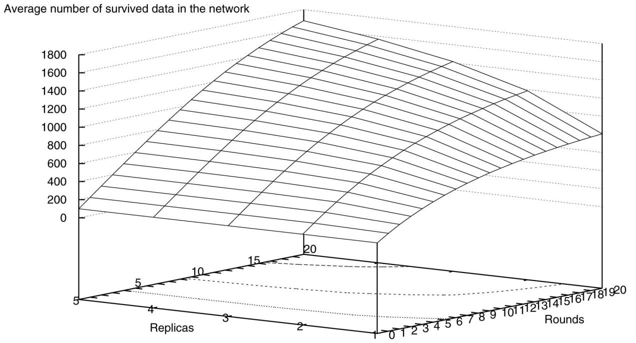

Fig. 6. Effects of replication against Eraser using the Frantic strategy; network strategy is KM.

with $R = 5, \mu$ADV deletes only 15 percent. This clearly represents a dramatic increase in data survival and confirms that KM is the best strategy when used with replication.

6 Related Work

Wireless Sensor Networks and Mobile Ad Hoc Networks have so many similarities that very often results of the one can be applied to the other. Even if researchers are mainly engaged to devise efficient routing or almost optimal clustering protocols, many results are also proposed for MANET data availability: those network face challenges such as communication faults, network partitions, or malicious node behaviors.

In those contexts, a research thread aims to buttress data availability to any MANET node, even in case of network fragmentation or congestion. Hara and Madria [17] introduced some simple and effective replication algorithms, such that even if a node lies in a disconnected partition, it can still access any data with high probability. A mechanism to provide replica consistency in case of updates to the original data and simple location management technique to guarantee that nodes access the closest replica are also presented in their work.

Data replication effectiveness in partitioned MANETs is also addressed by Gianuzzi et al. [14]. The authors show how data access in the network is highly related to the number of replicas as well as to the network density and to the nodes' transmission radius.

Chessa and Maestrini [3] introduced a distributed data storage approach for mobile wireless networks, based on the peer-to-peer paradigm. Their technique provides support to create and share files under a write-once model, and also ensures data confidentiality and dependability by encoding files in a Redundant Residue Number System (RRNS).

Benenson and coworkers [29] investigated possible strategies for preventing a mobile adversary from learning certain sensed data and/or for preventing contiguous unauthorized access, once the data have been learned. Data are randomly moved around the network and an adversary who once had access to the data stored at some captured sensor, must compromise other sensors in order to retain its access to the target data. Several algorithms are introduced to provide efficient data retrieval and update.

The authors of [1], [25], [4] statistically improve data confidentiality and data availability in hostile MANET environments, where both insider and outsider adversaries may be present, by leveraging the existence of multiple paths between end nodes.

A more recent result addressing data availability in WSNs is [18]. It develops a scheme to maximize the amount of data recovered at the sink and shows how the proposed scheme improves data availability when a portion of the network is invalidated by natural disasters, such as floods or earthquakes.

Our work is inspired by a seminal paper for UWSNs [8]. The authors introduce an adversarial model with limited scope and show how data survival can be achieved with noncryptographic techniques. The use of cryptography in UWSNs was recently addressed either to protect data confidentiality in [9] or to achieve data authentication, as in [10]. Collaborative approaches in UWSNs to cope with the mobile adversary can be found also in [23] and [7]. A broader analysis of the different mobile adversary's goals can be found in [21].

UWSNs have also recently been considered in the context of minimizing storage and bandwidth overhead due to data authentication in the presence of a powerful adversary [22]. The proposed forward-secure aggregate authentication techniques can efficiently provide *forward security*, i.e., having compromised a sensor, the adversary is unable to modify any data collected prior to compromise.

7 Conclusions

In this paper, we introduced a new mobile adversary model specific to unattended WSNs operating in hostile settings. We introduced some simple noncryptographic strategies for maximizing data survival in the presence of an adversary that has been characterized in different flavors, based on its goals. A thorough analysis completely frames the quality of the devised strategies, when applied to the different adversary models. Data dissemination and replication strategies described here demonstrate significantly improved probability of data survival. Finally, extensive simulations support our analytical findings.

The use of cryptography in the described framework is currently being explored.

Appendix A

Erasure Codes

As demonstrated earlier, replication is effective against Search-and-Erase and Eraser. However, replication results in a commensurate increase in storage and bandwidth costs.

One alternative is to use well-known erasure codes in order to improve data survival probability. To this end, in the rest of this section, we compare erasure codes against plain replication assuming the KM defense strategy. We adopt IDA (Information Dispersal Algorithm) [26] as the underlying erasure code, mainly because of its optimal bandwidth usage. Using IDA, it is possible to break an information in an arbitrary number of fragments, such that reconstructing the original information would only require a subset of the original fragments. Dispersal and reconstruction are computationally efficient, while a careful choice of the parameters can make dispersal also space efficient. The basic idea is to split each data item into $F = f' + f''$ fragments and treat them independently, offloading each of them to a random recipient. As long as f' out of F fragments survive, the sink can reconstruct the original value. Indeed, f'' characterizes the IDA reliability.

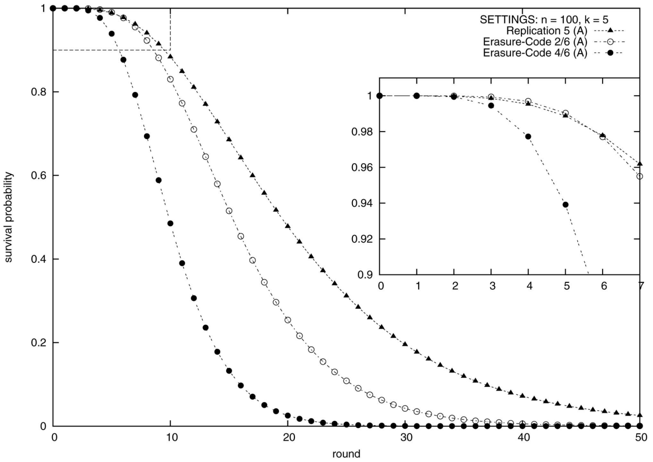

Fig. 7. Replication versus erasure codes. (A) stands for analytical results.

According to [26], a value is considered as a sequence of characters $b_1, \dots, b_N$. Let p be the smallest prime such that $b_i < p$. Then, each fragment is composed of $\frac{N}{f'}$ elements of Z_p. Thus, the number of bits to represent a value after fragmentation is $(f' + f'') \cdot \frac{N}{f'} \cdot log(p)$.

For a fair comparison, we need to use the number of bits for fragmentation and replication, i.e.:

$$R \cdot |x| = (f' + f'') \frac{N}{f'} \cdot log(p).$$

A.1 Search-and-Erase

A Search-and-Erase needs to delete at least $f'' + 1$ fragments of target data in order to succeed. The probability that μADV deletes one fragment by round r is $1 - p_{KM}(r)$.

If each data item is split into $f' + f''$ fragments, the probability that the value is irrecoverable is the same as the probability of *at least* $f'' + 1$ fragments being deleted by round r:

$$\bar{p}(r) = \sum_{i=f''+1}^{f'+f''} \binom{f' + f''}{i} (1 - p_{KM}(r))^i (p_{KM}(r)^{f'+f''-i}).$$

Thus, the probability of *at least* f' fragments surviving by round r is $p(r) = 1 - \bar{p}(r)$.

A.2 Comparison with Replication

We now compare fragmentation with plain replication. We assume that each data item is one byte. With replication and $R = 5$, the total number of bits is 40. With fragmentation, using the same number of bits, we consider two cases: 1) $f' = 2, f'' = 6$, and 2) $f' = 4, f'' = 6$.

As shown in Fig. 7, replication works better than fragmentation, except for the first few rounds. Indeed, up to $r = 5$, fragmentation performs (slightly) better than replication. This can be explained focusing on the first round: μADV can only win by deleting at least $f'' + 1$ fragments. Whereas, in case of replication, it has to delete exactly r items. However, the number of fragments is higher than the number of replicas. Therefore, as rounds go by, fragments are deleted with greater probability. As the number of fragments decreases, fragmentation starts loosing its appeal. We note that, with replication, as long as just one replica survives, μADV looses, while with fragmentation, at least f' fragments must survive. Fig. 7 shows that the advantage of fragmentation over replication vanishes after round 5.

We conclude that fragmentation does not seem worthwhile.

Acknowledgments

The authors are grateful to the anonymous referees for their insightful and useful comments on the earlier version of this paper. This research was supported in part by:

- The Spanish Ministry of Science and Education through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER CSD2007-00004 "ARES," and by the Government of Catalonia under grant 2005 SGR 00446.
- The EU FP6 projects: "European Satellite Communications Network of Excellence" (SatNEx II, IST-27393); PERSONA (Contract No. 045459); INTERMEDIA (Contract No. 38419).
- An award from the US Army Research Office (ARO) contract W911NF0410280 as well as a grant from the US Fulbright Foundation.

References

[1] V. Berman and B. Mukherjee, "Data Security in Manets Using Multipath Routing and Directional Transmission," *Proc. IEEE Int'l Conf. Comm. (ICC '06),* pp. 2322-2328, 2006.

[2] A. Caruso, A. Urpi, S. Chessa, and S. De, "GPS-Free Coordinate Assignment and Routing in Wireless Sensor Networks," *Proc. IEEE INFOCOM '05,* pp. 150-160, 2005.

[3] S. Chessa and P. Maestrini, "Dependable and Secure Data Storage and Retrieval in Mobile, Wireless Networks," *Proc. Int'l Conf. Dependable Systems and Networks (DSN '03),* pp. 207-216, 2003.

[4] M. Conti, R. Di Pietro, and L.V. Mancini, "ECCE: Enhanced Cooperative Channel Establishment for Secure Pair-Wise Communication in Wireless Sensor Networks," *Ad Hoc Networks,* vol. 5, no. 1, pp. 49-62, 2007.

[5] B. Deb, S. Bhatnagar, and B. Nath, "Reinform: Reliable Information Forwarding Using Multiple Paths in Sensor Networks," *Proc. 28th IEEE Int'l Conf. Local Computer Networks (LCN '03),* pp. 406-415, 2003.

[6] J. Deng, C. Hartung, R. Han, and S. Mishra, "A Practical Study of Transitory Master Key Establishment for Wireless Sensor Networks," *Proc. First Int'l Conf. Security and Privacy for Emerging Areas in Comm. Networks (SecureComm '05),* pp. 289-302, 2005.

[7] R. Di Pietro, D. Ma, G. Tsudik, and C. Soriente, "POSH: Proactive Co-Operative Self-Healing in Unattended Sensor Networks," *Proc. 27th IEEE Int'l Symp. Reliable Distributed Systems (SRDS' 08),* pp. 185-194, 2008.

[8] R. Di Pietro, L.V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Catch Me (If You Can): Data Survival in Unattended Sensor Networks," *Proc. Sixth Ann. IEEE Int'l Conf. Pervasive Computing and Comm., (PerCom '08),* pp. 185-194, 2008.

[9] R. Di Pietro, L.V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Playing Hide-and-Seek with a Focused Mobile Adversary in Unattended Sensor Networks," *Ad Hoc Networks,* special issue on privacy and security in wireless sensor and ad hoc networks, vol. 7, pp. 1463-1475, 2009.

[10] R. Di Pietro, C. Soriente, A. Spognardi, and G. Tsudik, "Collaborative Authentication in Unattended WSNS," *Proc. ACM Conf. Wireless Network Security (ACM WiSec '09),* 2009.

[11] Y. Frankel, P. Gemmell, P.D. MacKenzie, and M. Yung, "Proactive RSA," *Proc. 17th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '97),* pp. 440-454, 1997.

[12] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-Sync Protocol for Sensor Networks," *Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03),* pp. 138-149, 2003.

[13] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," *SIGMOBILE Mobile Computing and Comm. Rev.,* vol. 5, no. 4, pp. 11-25, 2001.

[14] V. Gianuzzi, "Data Replication Effectiveness in Mobile Ad-Hoc Networks," *Proc. ACM Workshop Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '04),* pp. 17-22, 2004.

[15] C.M. Grinstead and J.L. Snell, *Introduction to Probability,* second ed. Random House, 2003.

[16] M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, *Probabilistic Methods for Algorithmic Discrete Mathematics.* Springer-Verlag, 1998.

[17] T. Hara and S.K. Madria, "Data Replication for Improving Data Accessibility in Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 5, no. 11, pp. 1515-1532, Nov. 2006.

[18] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth Codes: Maximizing Sensor Network Data Persistence," *SIGCOMM Computer Comm. Rev.,* vol. 36, no. 4, pp. 255-266, 2006.

[19] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," *Ad Hoc Networks,* vol. 1, nos. 2/3, pp. 293-315, 2003.

[20] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. Sixth ACM/IEEE Ann. Int'l Conf. on Mobile Computing and Networking,* pp. 243-254, 2000.

[21] D. Ma, C. Soriente, and G. Tsudik, "New Adversary and New Threats: Security in Unattended Sensor Networks," *IEEE Network,* vol. 23, no. 2, pp. 43-48, Mar. 2009.

[22] D. Ma and G. Tsudik, "Extended Abstract: Forward-Secure Sequential Aggregate Authentication," *Proc. 2007 IEEE Symp. Security and Privacy (SP '07),* pp. 86-91, 2007.

[23] D. Ma and G. Tsudik, "DISH: Distributed Self-Healing in Unattended Sensor Networks," *Proc. 10th Int'l Symp. Stabilization, Safety, and Security of Distributed Systems (SSS' 08),* 2008.

[24] R. Ostrovsky and M. Yung, "How to Withstand Mobile Virus Attacks," *Proc. 10th ACM Symp. Principles of Distributed Computing (PODC '91),* pp. 51-59, 1991.

[25] P. Papadimitratos and Z.J. Haas, "Secure Data Communication in Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 2, pp. 343-356, Feb. 2006.

[26] M.O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," *J. ACM,* vol. 36, no. 2, pp. 335-348, 1989.

[27] T. Rabin, "A Simplified Approach to Threshold and Proactive RSA," *Proc. 18th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '98),* pp. 89-104, 1998.

[28] A. Shamir, "How to Share a Secret," *Comm. ACM,* vol. 22, no. 11, pp. 612-613, 1979.

[29] P.M. Cholewinski, Z. Benenson, and F.C. Freiling, "Simple Evasive Data Storage in Sensor Networks," *Proc. Int'l Conf. Parallel and Distributed Computing Systems (PDCS '05),* pp. 779-784, 2005.

Roberto Di Pietro received the Laurea degree in computer science from the University of Pisa, Italy, in 1994. He received the PhD degree in computer science from the Università di Roma "La Sapienza," Italy, in 2004. In 2004, he also received a Specialization Diploma in operating research and strategic decisions from the Department of Statistics of the same university. From 1995 to 2006 he served as an officer for the technical branch of the Italian Ministry of Defence. He has spent several periods abroad visiting the UNESCO Chair in Data Privacy, the Institut Eurecom, and George Mason University (Center for Secure Information Systems). He is currently an assistant professor of computer science in the Department of Mathematics of the Università di Roma Tre, Italy, and a research associate at the National Research Council, Pisa. His main research interests include security and privacy for RFID, mobile, ad hoc, and underwater wireless networks; intrusion detection; security and privacy for distributed systems; secure multicast; applied cryptography; computer forensics, and role mining for access control systems (RBAC).

Luigi V. Mancini received the Laurea degree in computer science from the University of Pisa, Italy, in 1983 and the PhD degree in computer science from the University of Newcastle upon Tyne, United Kingdom, in 1989. From 2000, he is a full professor of computer science at the Dipartimento di Informatica of the University of Rome "La Sapienza." Since 1994, he has been a visiting research professor of the Center for Secure Information Systems, George Mason University, Virginia. His current research interests include computer network and information security, secure multicast communication, public key infrastructure, authentication protocols, system survivability, computer privacy, wireless network security, fault-tolerant distributed systems, and large-scale peer-to-peer systems. He published more than 70 scientific papers in international conferences and journals such as the *ACM Transactions on Information and System Security*, *IEEE Transactions on Knowledge & Data Engineering*, *IEEE Transactions on Parallel & Distributed Systems*, and *IEEE Transactions on Software Engineering*. He served on the program committees of several international conferences, including the ACM Conference on Computer and Communication Security, the ACM Conference on Conceptual Modeling, the ACM Symposium on Access Control Models and Technology, the ACM Workshop of Security of Ad-Hoc and Sensor Networks, and IEEE Securecomm. He is the guest editor of the special issue hot topics in peer-to-peer systems of the journal *Concurrency and Computation: Practice and Experience,* April 2008. He is the founder of the Information and Communication Security (ICSecurity) Laboratory, see http://icsecurity.di.uniroma1.it. Currently, he is a member of the Scientific Board of the Italian Communication Police Force and the director of the Master Degree Program in Information and Network Security of the University of Rome "La Sapienza."

Claudio Soriente is currently a PhD candidate in the Donald Bren School of Information and Computer Science at the University of California, Irvine. His research interests include network security, wireless sensor networks, and usable security.

Angelo Spognardi received the PhD degree in 2008 from the Dipartimento di Informatica of the Università degli studi di Roma "La Sapienza," under the tutoring of Professor Luigi V. Mancini. He is currently a postdoctoral researcher at INRIA Rhône-Alpes, with the Plantè Equipe of Claude Castelluccia. His main research interests include security and privacy for RFID, wireless, and wireless sensor network security; intrusion detection; security and privacy; and cryptography.

Gene Tsudik received the PhD degree in computer science from the University of Southern California in 1991 for research on firewalls and Internet access control. He is a professor of computer science and the director of the Secure Computing and Networking Center (SCONCE) at the University of California, Irvine (UCI). He has been conducting research in internetworking, network security, and applied cryptography since 1987. Before coming to UCI in 2000, he was a project leader at the IBM Zurich Research Laboratory (1991-1996) and the USC Information Science Institute (1996-2000). Over the years, his research interests included routing, firewalls, authentication, mobile networks, e-commerce, anonymity, group communication, digital signatures, key management, ad hoc networks, as well as database privacy and secure storage. Between 2003 and 2007, he was an associate dean of research and graduate studies in the School of Information and Computer Sciences at UCI. He spent April-September 2007 in Italy as a Fulbright Scholar at the University of Rome "La Sapienza." He currently serves as a vice-chair of graduate studies in the Computer Science Department at UCI.