Imitation Assault Detection in a Region Partitional Distributed Approach for a Wireless Sensor Network

Madhumathi Rajesh
Meenakshi College of Engineering,Chennai

Rama Sugavanam
Meenakshi College of Engineering,Chennai

G R Gangadevi
Meenakshi College of Engineering,Chennai

ABSTRACT
The vital problem over the Wireless Sensor Networks (WSNs) are that they are often vulnerable to attacks where an adversary can easily compromise some of the nodes, can reprogram, and then, can imitate them in a large number. They distribute the clones in the network, launching node replication attacks or clone attacks by loading secret information into several replicated nodes and rejoining these nodes to execute malicious behaviors or threaten underlying protocols. Earlier works against clone attacks suffer from either a high storage or poor detection accuracy. In this paper we are proposing a new remedial, algorithm called RERD (Region based – Efficient, Randomized, and distributed) that detects the clone attack achieving a higher probability of detection. The wireless zone is partitioned into regions with the new DRCS algorithm followed by clone detection using TWG algorithm which is a combination of Token message and witness node.

Keywords
WSN, clone attack, region,RERD,distributed.

1. INTRODUCTION
Wireless sensor networks (WSN) enable simultaneous, high-speed sensing and data acquisition such as temperature, pressure, position, flow, humidity, vibration, biomedical, force and motion. Sensor nodes are cheap, resource limited sensing devices which can communicate at short distances, and have a small amount of memory and computing power. In sensor networks, a rival may easily capture and compromise sensors and deploy unlimited number of clones of the compromised nodes. Since these clones have genuine access to the network, they can participate in the network operations in the same way as a legitimate node, and thus launch a large variety of insider attacks [1, 2, 3], or even take over the network. If these clones are left undetected, the network is unshielded to attackers and thus extremely vulnerable. Therefore, clone attackers are severely destructive and hence efficient solutions for clone attack detection are needed to limit their damage. In this paper, we propose a novel scheme for detecting clone attacks in sensor networks with a new region based efficient, randomized, and distributed (RERD) algorithm. Our algorithm refreshes in a regular clock period thereby improving the node replication detection rate and also we prove that our protocol is self healing by nature. The rest of this paper is organized as follows: Section 3 shows reviews related work in both centralized and distributed approach; Section 4 shows the RERD threat model

assumed in this paper; Section 5 describes our region based efficient, randomized, and distributed (RERD) algorithm; Section 6 gives some simulation results and Section 7 presents some concluding remarks.

2. RELATED WORKS

2.1 Centralized Approach
A straightforward solution to defend against clone attacks is to let the base station collect the neighborhood information (id, Location) from each sensor and monitor the network in a centralized way. This approach suffers from high communication overhead by requesting redundant information from the network. Another centralized clone detection protocol has been proposed in [5]. This solution assumes that a random key pre distribution security scheme is implemented in the sensor network. That is, each node is assigned a set of k symmetric keys, randomly selected from a larger pool of keys [6]. For the detection, each node constructs a counting Bloom filter from the keys it uses for communication. Then, each node sends its own filter to the BS. From all the reports, the BS counts the number of times each key is used in the network. The keys used too often (above a threshold) are considered cloned and a corresponding revocation procedure is raised.In other solution, a localized voting/misbehavior detection where nodes within a neighborhood agree/vote on the legitimacy of a given node based on their local observations. Nevertheless, these schemes are not capable of detecting clones with normal behavior, and may fail when multiple clones in close proximity collude. Furthermore, localized voting/misbehavior detection schemes inherently lack the ability to detect distributed clones that may appear at any place in the network In one-hop networks, the base station (BS) can store the unique signal characteristic for each device, and thus device cloning can be detected accordingly. However, in a multi-hop sensor network, it is impractical for BS to track the signal characteristics of sensors multi-hops away.

2.2 Distributed Approach
A naive distributed solution for the detection of the node replication attack is Node-To-Network Broadcasting. In this solution, each node floods the network with a message containing its location information and compares the received location information with that of its neighbors. If a neighbor Sw of node Sa receives a location claim that the same node Sa is in a position not coherent with the originally detected position of Sa, this will result in a clone detection. However, this method is very energy-consuming since it requires n flooding per iteration, where n is the number of nodes in the WSN.Another distributed solution is to detect clones based on set operations. In [7], Choi et al. propose to divide a sensor network into exclusive sub regions and check if there is any overlapping between them. A non-empty intersection indicates the existence of replicated sensors. The results of the membership checking are united and authenticated along a tree structure, and sent to the base station finally. Despite the fact that the number of messages is reduced to O(N), the length of the messages increases linearly, and the total amount

of data to be transferred for membership checking is not reduced at all.Parno et al. proposed two emergent protocols based on the distributed verification of the location claims. These distributed schemes are based on passive discovery of the replicated nodes by witness nodes storing signed locations claims. The first one, Randomized Multicast (RM), distributes node location information to randomly selected nodes. The second one, Line-Selected Multicast (LSM), uses the routing topology of the network to detect replicas. In RM, when a node locally broadcasts its location, each of its neighbors sends with probability p, a digitally signed copy of the location claim to a set of randomly selected nodes. Assuming that there is a replicated node, if every neighbor randomly selects Þ destinations, with a not negligible probability, at least one node will receive a pair of not coherent location claims. We will call witness the node that detects the existence of a node in two different locations within the same protocol run. The RM protocol implies a high communication cost: Each neighbor has to messages. The LSM protocol is similar to RM, but it introduces a remarkable improvement in terms of detection probability. In LSM, when a node announces its location, every neighbor first locally checks the signature of the claim, and then, with probability p, forwards it to g - 1 randomly selected destination nodes.The basic idea is to logically divide the network into cells and to consider all the nodes within a cell as possible witnesses.In the first proposed protocol, Single Deterministic Cell, each node ID is associated with a single cell within the network. When the protocol runs, the neighbors of a node a probabilistically send a's claim to the single predetermined witness cell for a. Once the first node within that cell receives the claim message, the message is flooded to all the other nodes within the cell.In the second proposal, Parallel Multiple Probabilistic Cells, the neighbors of a node a probabilistically send a's claim to a subset of the predefined witness cells for a. The proposed solutions show a higher detection probability compared to LSM. However, the same predictable mechanism Used to increase the detection probability can be exploited by the adversary for an attack—compromising the witnesses in order to go undetected. In fact, this predictability restricts the number of nodes (and their geographic areas) that can act as witnesses.A randomized, efficient, and distributed clone detection protocol (RED protocol) which is similar in principle, to the Randomized Multicast protocol [8], but with witnesses chosen pseudo randomly based on a network-wide seed. RED achieves a large improvement over RM in terms of communication and computation. When compared with LSM [8], a protocol that is more efficient than RM, RED proves to be again considerably more energy efficient.RED executes routinely at fixed intervals of time. Every run of the protocol consists of two steps. In the first step, a random value, rand, is shared among all the nodes. This random value can be broadcasted with centralized mechanism, or with in-network in distributed mechanisms. A secure, verifiable leader election mechanism [9] can be used to elect a leader among the nodes; the leader will later choose and broadcast the random value. In the second step, each node digitally signs and locally broadcasts its claim—ID and geographic location. When the neighbors receive the local broadcast, they send with the probability p, the claim to a set of g _ 1 pseudo randomly selected network locations.For every genuine message witness node extracts the information (ID and location). If this is the first claim carrying this ID, then the node simply stores the message. If another claim from the same ID has been received, the node checks if the new claim is coherent with the claim stored in memory for this ID. If it is not, the witness declares the two incoherent signed claims are the proof of cloning.Adding more efficiency to the RED protocol we have planned to concentrate on witness selection and distribution based on regional partition. RERD is an extension of RED protocol proposed in [10], where the wireless zone is subdivided into regions on the time basis and with the roaming token the witness is chosen. Further clone detection process is preceded with the claim transaction between the nodes and the witness.

3. RERD – THREAT MODEL

The complete wireless sensor precinct is partitioned into sub regions on the basis of time slots with the region constructor algorithm briefed in the Section 5. A token (id, rand) is set to spin inside each of the region. The sensor that gets the token at that particular spark of time, will act as the witness node.The witness node gathers the claim (id, loc) from all the other sensors located in that particular region. It generates a special table called status-Table which checks for the presence of any clones (Intra - Region). Two nodes with the same id but with the different locations will be identified as clones.Once this process is over, all the witness nodes will transfer the state_table table that holds the id's of the nodes present in their respective regions to the base station (BS). The BS will merge all the State_Tables with the master_Table and routinely check for any inconsistencies of id's thereby clones present inter – region basis will also be identified. The BS will run a periodical event handler that takes care of region construction, witness selection and Inter – Intra Clone detection. As the process is repeated sporadically new regions are constructed with fresh witness nodes and with new revitalized table entries. This is a good indication that the intruders should be careful enough to get trapped. This process greatly condenses the clone attacks.The Table 1 narrates the inter clone attack scenario with its master_table entries. The presence of ID5a in region A and B shows the presence of clones in the regions. Similarly the state_table entries present in the witness node determines the replicas among them.

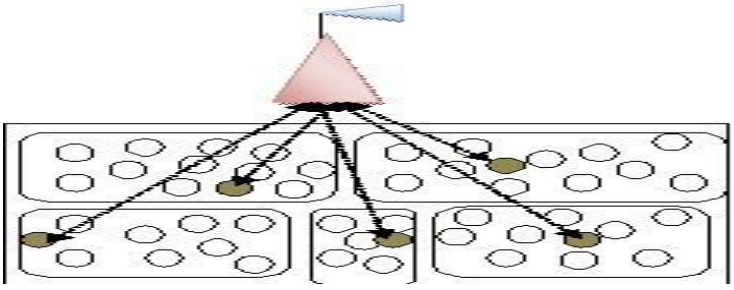

Fig 1. WSN with Region & Witness.

Table 1 : Base Station – Inter Clone attack Detection (master_Table)

Region	Time	ID – List
A	T_1	ID_{1a}, ID_{2a}, ID_{3a}, ID_{4a}, ID_{5a}, ID_{6a}, ID_{7a}, ID_{8a}, ID_{9a}, ID_{10a}, ID_{11a}, ID_{12a}
B	T_1	ID_{1b}, ID_{2b}, ID_{3b}, ID_{4b}, ID_{5a}, ID_{6b}, ID_{7b}, ID_{8b}, ID_{9b}, ID_{10b}, ID_{11b}, ID_{12b}
C	T_1	ID_{1c}, ID_{2c}, ID_{3c}, ID_{4c}, ID_{5c}, ID_{6e}, ID_{7b}, ID_{8c}, ID_{9c}
D	T_1	ID_{1d}, ID_{2d}, ID_{3d}, ID_{4d}, ID_{5d}, ID_{6d}
E	T_1	ID_{1e}, ID_{2e}, ID_{3e}, ID_{4e}, ID_{5e}, ID_{6e}, ID_{7e}, ID_{8e}, ID_{9e}, ID_{10e}

4. RERD ALGORITHM

RERD, Region based Efficient Randomized Algorithm is split into two modules where it is the combination of DRCS (Distributed Region Construction Scheme) and TWG (Token – Witness Generation) algorithms. The subsequent sections enumerate the architecture of the algorithms in detail.

4.1 Region Construction: DRCS

A distributed region construction scheme (DRCS) for periodical data gathering is proposed in this paper. The region heads (RH) are elected from a number of candidate nodes with the node residual energy and node density nearby.

The scheme is fully localized and produces an even distribution of regions approximately. In the region construction phase, the plain nodes choose to join the region according to both distance and load balance.The overall region head selection takes place in the following scenario. The base station broadcasts a greet_MSG to the sensors in the network. Polling among the sensors that compete for the region head is performed.

A compete_ME_ MSG in turn is broadcasted as long as the head is elected. The winner among them will be decided as RH.Once the greet_MSG is received by various nodes located globally, they become the competer for the region head. Each of the competer broadcasts the compete_ME_MSG to its neighbor. The node that acts as the competer will be called as the βnode. The neighbor who receives the compete_ME_MSG will try to win the βnode with its energy level. The node with the highest energy level in each of the surrounding till all the nodes have been traversed as long as the time is out will be named as the RH. Thus region heads are allotted in different geographic positions.Later, all the RH's broadcasts friend_REQ message to its neighboring nodes. In turn the neighbors acknowledge them with friend_ack message based on their load level.

If positive acknowledgement is received it indicates that the node is prepared to join that particular region. Else a negative acknowledgement is transmitted to the node which concludes that the node is not ready to become the part of this region. The process is repeated as long as all the nodes are identified. Every node will be traversed and it becomes the part of any of the neighboring region. No two regions will have a node of same id. If a node receives request from more than one region head then based the energy levels the node will decide to join in any of the region sending a negative acknowledgement to the rest of the other region heads. Some nodes with less energy may respond with negative acknowledgement, those will form separate regions. Thereby a number of regions are constructed geographically.

The DRCS algorithm narrates the above courses of action with the following two procedures.

- REGION_HEAD selection.
- REGION_FORMATION.

In the first procedure, region heads are selected with the node holding the high residual energy. The second procedure forms the regions periodically. Finally after the formation of the region's by the DRCS algorithm the paper concentrates to the next phase with the clone detection.

4.2 Token Manupulation and Witness Selection: TWG

TWG, Token – Witness Generation algorithm takes care of clone detection once after the region formations are accomplished. Tokens are special messages generated by RH's which is a combination of a unique transaction id and a random number.

These tokens are broadcasted by all the RH's to the randomly selected node in each of their respective regions. The token message is passed among the rest of the nodes located across every region.

Once the BS transmits a witness node identification message, those nodes within each of the region with the token message at that particular instance will report as the witness node of that corresponding region. Thus various witness nodes are generated and are reported to the base station. This process is periodically repeated for every new region formations.

4.3 Intra – Clone Dedection

The witness node broadcasts a request to the rest of the nodes within its region for a *claim message*. In turn all the nodes within the region respond the witness node with the ***claim_reply*** which is a combination of id and the present geographic location of that particular node. The transaction is embedded in a digitally signed key cryptographic exchange The witness node accumulates the gathered information from each of the neighboring nodes within every region and maintains the same in a ***status_table***.

The table holds the id's and locations of every individual nodes present in its region. Later the algorithm explores for any two similar id's but with distinct locations among the table entries. On such a circumstances ***clone_alert*** message will be genetated. This message of discrepancy will be broadcasted to the BS informing the intra - clone attacks. The procedure is repeated in every region by the witness node. Once the process of gathering and examination is completed the entries of the ***status_table*** is updated to the ***master_table***.

Algorithm 1: DRCS

```
Procedure REGION_HEAD selection
    BS randomly broadcasts "GREET_MSG"
    ∀ nodes that receive "GREET_MSG" do
        if (Is_GREET_MSG(θ_node))
            Poll θ as COMPETER
            θ broadcasts "COMPETE_ME_MSG" to neighbor()
          ∀ neighbor() do
            if (θ_energy <NB_energy)
                θ=Swap(θ,NB)
            endif
            select next NB_node
          while (Time out)
        return θ_state ← RH_r
        endif
    while all the nodes are identified
    call procedure REGION_FORMATION
end procedure
Procedure REGION_FORMATION
    RH_x: Region Head of different location
    R: Starting Region
    Rμ:New Region
    R=R ∪ RH_x
    ∀ nodes y do
        RH broadcasts "FRIEND_REQ"
        if (y_load < RH_threshold)
            reply "Friend_ack"
            R=R ∪ y "Join the region"
        else if (free node) then
            Reply "Friend_Nack"
            Wait for the next request "join the new region"
        else
            Rμ = Rμ ∪ { ε } "Create a new region"
        endif
    end for
end procedure
```

Algorithm 2: TWG

```
RH_x generates Tokens
  procedure Token_Manupulation
    Token ← (id_Rx,region(),rand(),timer())
    Digitally_Signed_Token ← (Token, K_BS^Private (Token))
    RH ← neighbor() (id_RH,neighbor(),Digitally_Signed_Token)
    Witness ← Digitally_Signed_Token
  end procedure
  procedure Intra_Clone_detect
    Claim ← (id_x, location(), time())
    Encrypt_claim ← (Claim, K_wt^private(Claim))
    Witness broadcast Claim_REQ
    ∀ nodes within the region do
      if(is_Claim(x))
        Encrypt_claim ← (Claim,signature)
        Update status_table
      else
        Discard claim
      end if
    end for
    ∀ entries in the table do
      Check for inconsistent_claim
      if(is_inconsistant(claim_x))
        Trigger exception "clone_alert"
      else
        Broadcast status_table to BS
      end if
    end for
  end procedure
  procedure Inter_Clone_detect
    ∀ status_table received do
      Update master_table
      if(is_inconsistant_id_entry)
        Trigger exception "clone_alert"
      else
        Call REGION_HEAD procedure
      end if
    end for
  end procedure
```

4.4 Inter – Clone Attack Detection

The BS broadcasts a status_update command to all the witness nodes for which the witness nodes will reply with the status_table to the BS. The BS consolidates the entries sent by the witness node, and revises the master_Table. Now the Base Station will check for replicated information. Any repeated entries in the master_Table determine data inconsistency. This shows the presence of clones among the regions.The procedure Token_Manupulation takes care of token generation and witness node identification. The method Inter_Clone_Dectect determines the inter clone attacks among the nodes within the regions and the routine Intra_Clone_Detect discovers the clone attack it the larger basis. The Algorithm 2 briefs the Token manipulation; inter clone and intra clone attack.

5. SIMULATION AND RESULTS

In this section we evaluate our algorithms by simulations. In our simulations, we randomly deploy 10000 nodes within a 1000m×1000m square. The transmission range is set to 50m. Also we test our protocols in a variety of irregular network topologies. We assume occasional packet losses can be solved by retransmission mechanisms in lower layer protocols. The figure 2 shows the detection probability (y-axis) at different protocol iterations (x-axis). In particular, we plotted the detection probability for the first 200 runs. Plotted values were computed averaging the results obtained for 10000 network deployments. For all the considered iterations, the RERD protocol shows a better detection probability. More than one node can witness a clone attack; compromising a witness node does not imply that a clone attack will go undetected for RERD.

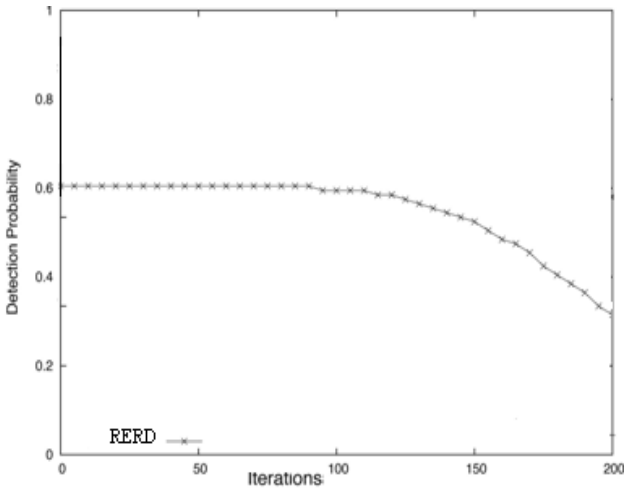

Fig 2. Simulation results

6. CONCLUSION

In this paper we have proposed a new region based duplicate detection approach (RERD) for a distributed environment. Our algorithm achieves higher degree of clone detection that is perilous, based on region distribution. The region formation and clone detection are periodically accomplished and hence even if some nodes are compromised they will get trapped by the algorithm. In future we would like to do more experiments improving the efficiency of the algorithm and decreasing the overhead cost.

7. REFERENCES

[1] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In IEEE Infocom'05, 2005.

[2] F. Liu, X. Cheng, and D. Chen. Insider attacker detection in wireless sensor networks. In IEEE INFOCOM'07, Anchorage, Alaska, 2007.

[3] Y. Yang, X. Wang, S. Zhu, and G. Cao. Sdap: a secure hopby-hop data aggregation protocol for sensor networks. In MobiHoc '06, pages 356–367, 2006.

[4] M. Conti, R. D. Pieto, and L. V. Mancini. "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks", In Proceedings of the 8th ACM International Symposium on mobile Ad Hoc Networking and Computing (MobiHoc) ,2007.

[5] R. Brooks, P. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M.T. Kandemir, "On the Detection of Clones in Sensor Networks Using Random Key Predistribution," IEEE ransSystems, Man and Cybernetics, Part C: Applications and Rev., vol. 37, no. 6, pp. 1246-1258, Nov. 2007.

[6] L. Eschenauer and V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," Proc. Conf. Computer and Comm.Security (CCS '02), pp. 41-47, 2002.

[7] H. Choi, S. Zhu, and T. Laporta. Set: Detecting node clones in sensor networks. In SecureComm'07, 2007.

[8] B. Parno, A. Perrig, and V.D. Gligor, "Distributed detection of Node Replication Attacks in Sensor Networks," Proc. IEEE Symp. Security and Privacy (S&P '05), pp. 49-63, 2005.

[9] G. Chen, J.W. Branch, and B.K. Szymanski, "Local Leader Election, Signal Strength Aware Flooding, and Routeless Routing," Proc. IEEE Int'l Parallel and Distributed Processing Symp.(IPDPS '05), p. 244.1, 2005.

[10] Mauro Conti, Roberto Di Pietro, Luigi Vincenzo Mancini, and Alessandro Mei, "Distributed Detection of Clone Attacks in Wireless Sensor Networks" IEEE transactions on dependable and secure computing, vol. 8, no. 5, september/october 2011

[11] H´enoc Soud´e, Jean M´ehat , "Energy Efficient Clustering Algorithm forWireless Sensor Networks".