Improving the representation of communities and their relations using local algorithms for community detection along side Security of low cost Radio Frequency Identification.

Michael Kizito

Student number: s0535265

May 05, 2008

Abstract

This research was done in two parts. The first part was done in Uganda and the second part was done in Netherlands. The research was done in different areas and is therefore not interlinked as one may suppose. The research done in Netherlands was based on low cost RFIDs and their security while the research done in Uganda was about representation of communities and their relations using local algorithms for community detection. In this brief we make mention of the work that has been done in both of the parts. Firstly we give a summary of what has been done in the first part;

RFID technology is steadily getting applied in pervasive computing. RFID systems are basically made up of tags, a reader and a backend database to which the reader is connected. In future RFID tags may replace the currently used barcodes (Universal Product Code) as they seem to offer much better functionality in comparison to barcodes.

In our study we explore the available RFID authentication protocols that have been proposed. We give describe how these protocols do the authentication with a small amount of rewritable memory and limited computing ability. The suggested protocols indicate that standard cryptography is not a prerequisite for improving security in RFID tags. Our contribution is to show how the RFID authentication protocols achieve the security properties. The security properties we consider are complexity, privacy, ability to protect against clone attacks, DoS attacks and forward security.

After the summary on RFIDs we go on to give a brief on the community detection as you shall read;

A community has a number of varying definitions but we shall at define it in two ways. Firstly a community is a subset of nodes on the network such that nodes in the same community are more likely to be connected than nodes in different communities. Examples could be division of social networks in groups, division of biological networks, routing in communication networks and so on.

There are algorithms that are used for community detection and these are the ones we intend to look at in this thesis and later on come up with an improved way of how to represent communities and their relations. We shall look at various networks ranging from social networks to biological networks so as to come up with a more appropriate way of community detection.

Course	Informatica/ Computer Science
Research Project	Studying community detection with the help of local algorithms along side Security in low cost RFIDs
Faculty	Faculty of Science
Research Institute	Institute for Computing and In formation Sciences (ICIS)
Research Group	Information and Knowledge Systems and Security of Systems Group
University	Radboud University Nijmegen
Supervisors	Prof. dr. Theo van der Weide dr. Peter van Rossum

Declaration:

I, Michael Kizito hereby declare that this thesis is my own work and has never been used before in any institution.

It concludes the research for my Masters degree and was carried out at the Information Retrieval and Information Systems research group and Security of Systems Group, under the supervision of Prof.dr.Theo van der Weide and dr. Peter van Rossum.

Signature ....

Date ....

Supervisor: Prof.dr.Theo van der Weide

Signature ....

Date ....

Supervisor: dr. Peter van Rossum

Signature ....

Date ....

Dedication

This work is dedicated to the Lord Jesus Christ and I would like to thank Him for His faithfulness and for the strength to get through this Masters course.

Jeremiah 29:11

Acknowledgement

Sincere thanks go to my supervisor Prof. dr. Theo van der Weide for all the support and guidance throughout the course. Thanks for the parental guidance and support.

Thanks also go to Joseph Martin Kavuma and all the colleagues we were with under the Nuffic I project for all the support.

Thanks go to Nicole, Paula, Marijke and the International relations staff at Student Administration who were always encouraging and supporting us throughout the course.

Thanks go to my family and my wife now for all the moral and spiritual support.

Thanks go to all the members of the Studenten kerk at Nijmegen and the members of the house group for all the support and encouragement.

1. Introduction of Part One

In this day and age many people have resorted to the World Wide Web to make themselves known or as a means of disseminating information about a given topic or subject.
Individuals who choose to publish information about themselves do so by using web pages that are referred to as Weblogs or Blogs. Since many of these individual weblogs are for particular people it is such a big task to identify a set of weblogs that form a natural group simply because the content may differ considerable for social reasons. Furthermore, social influence can be used to explain the behaviour of individuals by their thoughts, feelings and actions either directly or indirectly.

Some traditional methods have been devised as a means of findings communities and they focus exclusively on topology analysis. The Web has gained interest in the scientific community and this has resulted in various studies concerning a wide variety of topics.
Researchers in this area look at the structural properties of the World Wide Web. The contents of the websites are not of much importance and therefore the Web is seen as a graph.

A community has a number of varying definitions but we shall at define it in two ways. Firstly a community is a subset of nodes on the network such that nodes in the same community are more likely to be connected than nodes in different communities. An illustration of a community is in Figure 1. Examples could be division of social networks in groups, division of biological networks, routing in communication networks and so on.

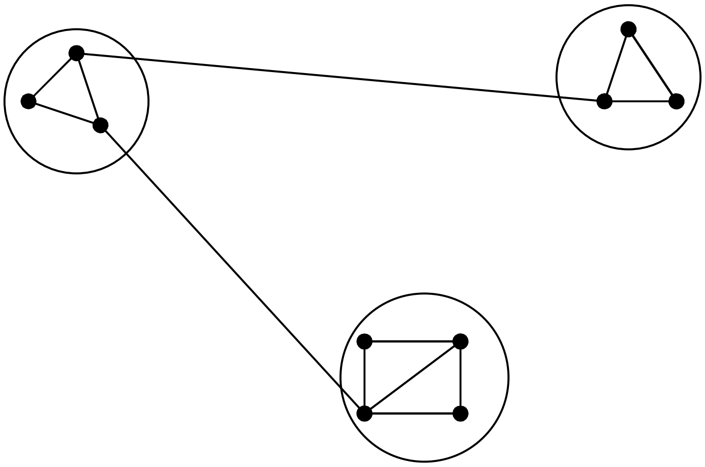

Figure 1 Pictorial representation of a community

1.1 Web Graph Theory

There are millions of HTML pages in the Net and most of them contain links. These links normally have a direction. Some pages receive much attention and have many pages referring to them: software repositories, international organizations, pages written by acknowledged experts in various fields etc. Conversely, there are pages with many links, such as resource lists. Some clever people realized that they could build better search engines by relating the two types that is the authority and the librarian as illustrated below:

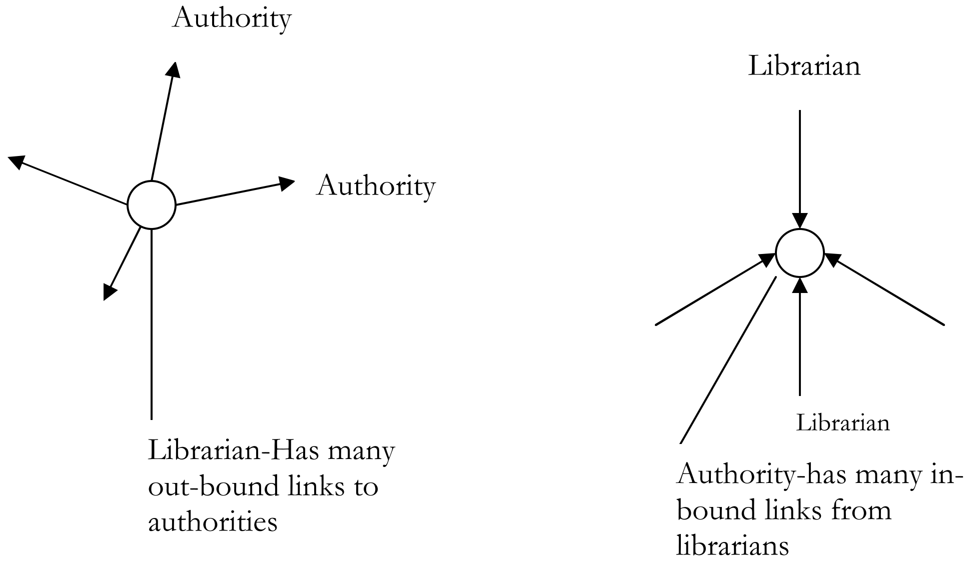
Authority
Authority
Librarian-Has many out-bound links to authorities
Librarian
Librarian
Authority-has many in-bound links from librarians

Figure 2 Illustrating the idea of an Authority and a Librarian

The above is an all encompassing definition but it works quite well. Iterative algorithms can take usual text-search results, thereby sorting out things and producing the "authorities". The commonly used search engine Google does something similar. Any extension of these ideas can lead to finding concepts and applications of Graph Theory in the Web map.
The Web is a huge directed graph G= (V, E) where the set V of vertices is the set of all pages and the set E of arcs corresponding to all pointers. Looking at the figure 2 the page X points to Y and Z and is pointed to by pages W and U. Vertex X has both outdegree and indegree equal to 2.

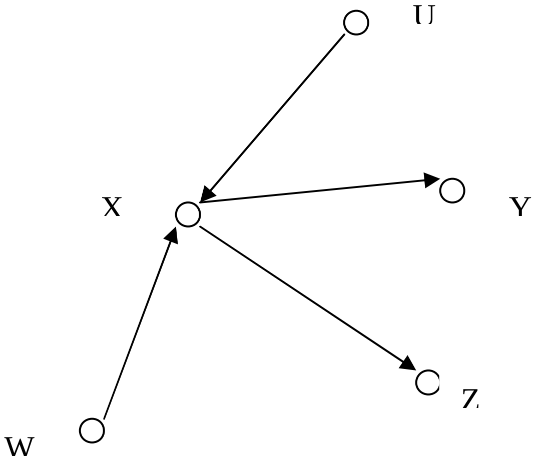

Figure 3 illustrates the indegree and outdegree of vertex X

It has been noted that there is no link between two pages and so we can say that the Web is not a connected graph. It is rather a sparse graph. The pointers are much less in comparison to the maximum possible. One may wonder why we say that the Web is not a connected graph and yet from the definition of the Web almost every page is accessible to anybody. Our argument comes from the idea that everything is reachable from everywhere just by links. Furthermore there are isolated groups of pages with no link from or to the outside world. There can be isolated vertices not pointing to or being pointed to by anything. Examples of some these can be personal pages with just a CV, without pointers and only accessible only from their known URL.

After giving some little information about the Web we shall from now on stay within a subgraph G' of the Web. By this we mean some subset of the Web pages together with the arcs that have endpoints in this subset. The subset could be a "community" of pages on a particular topic, or all commercial pages in some domain. When we have four pages each pointing to or being pointed to by others, we say we have a complete subgraph in the Web. For instance, "if there is a rule that if I point to you and you point to P, then I will also put a link to P." As a result we have complete sub graphs called tournaments with a pattern in the in-outdegrees of the vertices. We have an illustration in figure 3.

A minimal subset of vertices from which every other vertex in G' can be reached immediately is called a minimal dominating set of G'. This can be taken as an economic collection of bookmarks. We now arrive at the dominance number which is the number of vertices in a minimal dominating set of G'. In a specialized community with "outgoing" members, it is most probable that there are various sequences of pages each pointing to the next one, starting at P and ending at Q. If we take the arcs to have lengths of one, the shortest path should be the sequence with the minimum number of steps. Arcs may have varying lengths for example the distance between the hosting servers. The distance could be in terms of geographical distance or connection time and many more. The shortest path would approximate the fastest way to locate page Y starting from X. People have come up with a unique way of getting from every page to another. There is also the root which is not a page of particular importance per se but it is the best location to start building the optimal tree. Furthermore we can say that the root is not a resource list as it can point to only one other page for instance. In the collection of arcs, we cannot go back to the starting page and this comes from the definition of the tree.

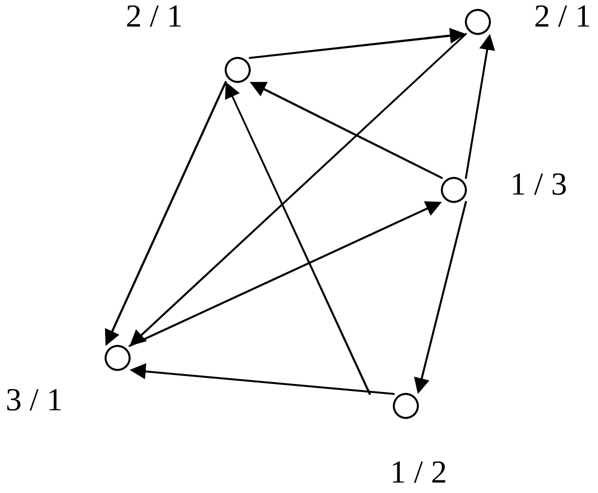

Figure 4 A pattern of in-outdegree of vertices

As a way of building a bookmark list, we consider a time of fixed length say L. Each one of the non-bookmarked page has some connection-based measure from the list and this measure is taken from the nearest bookmarked page. In order to cover all the community in such a way as to minimize the sum of all these connection based measures (distances), we are in essence solving the minisum problem otherwise known as the L-Median problem. In practise this problem is seen as a way of installing facilities in an optimal way for customers, and this would mean locating of mirror sites.

On the other hand we have the L-Centres problem where we try to minimize the maximum connection based measure. It is slightly the same as the facilities location problem but here the idea is more or less avoiding extreme delays in serving individual customers rather than the community as a whole.

Many algorithms have been proposed as a means of discovering communities; however it is hard to quantify performance given the fact that we clearly want the “best” algorithm.
There are some ways that have been used or tried out in the attempt to determine the “best” algorithm. These include

- The “Four groups” test for performance
- Bayesian inference problems
- Error-correcting codes
- The Belief Propagation (BP) and MFT algorithms for community detection
- Outlook

The “four groups” test starts with N nodes and they get divided into q different communities typically N = 128 and q = 4. After the division, the nodes are connected in the same communities independently at random with probability P_{out}. An average number of links can be defined as Z_{in} and Z_{out} for the same and different community respectively.

For the Bayesian method it advances that for a given community assignment to be correct is proportional to the prior expectation for that assignment multiplied by the probability that the given community assignment produces the given graph.
This can be further explained as follows
Let q_i be the community assignment for node i where $1 \leq q_i \leq q$
So the probability of producing a given graph is equal to

$$p(\{q_i\}) \exp [\sum J \delta q_i , q_j] \exp[\sum J' \delta q_i , q_j / 2]$$

$$J = \log [(P_{in} (1- P_{out})) / P_{out} (1- P_{in})],$$

$$J' = \log [(1- P_{in})) / (1- P_{out})]$$

As a means of simplifying matters, we ignore the prior knowledge of exactly 128/4 = 32 nodes in each community and just use p ($\{q_i\}$).
Considering the artificial graph already advanced by Girvan and Newman, we realise that it is a simple network with N nodes divided into four groups. Connections between pairs within a group are present with probability $p_{in,}$ while pairs of nodes in different groups are connected with probability $p_{out.}$ As the probability p_{out} grows from zero, the community structure in the network becomes less well defined.
Community structure identification has created a great interest among physics and computer society who are focusing on the properties of complex networks like the Internet, social networks, citation networks and email networks. A complex network is a representation of a complex system from real life in terms of nodes and edges, where a node is an individual member in the system and an edge is a link between nodes according to relation in the system. [40] One property of complex networks community structure can be described as the gathering of vertices into groups such that there is a higher density of edges within groups than between them. With the preceding definition the nodes in the community should have more intra-community connections rather than inter-community connections. Many methods and algorithms proposed reveal the underlying community structure in complex networks. The algorithms require the definition of community that imposes the limit up to which a group should be considered a community.
The success of a community detection algorithm squarely depends on how it defines a community. Girvan and Newman proposed a network modularity which has been used up widely in recent studies as the quality metric for assessment of partitioning a network into communities.

$$Q = \sum_i (eii - ai^2)$$

In the above equation *i* is the index of the communities, e_{ii} is the fraction of edges, that connects two nodes inside the community *i*, to the total number of edges in the network and
a_i is the community *i* to the total number of edges in the network. Network modularity calculations need less computational time when compared to edge betweenness centrality used in Girvan-Newman algorithm. This makes network modularity a practical measure to use it is large networks.

1.2 Belief Propagation (BP)

This one makes use of the same equations on a graph with a low density of loops. It does not give exact values but at least a good approximation.
Belief propagation is also known as the sum-product algorithm—an iterative algorithm for computing marginals of functions on a graphical model. This algorithm was formulated on trees by Judea Pearl in 1982 [41] and in 1983 it was further formulated on poly trees by Pearl and Kim [42]. This algorithm is an efficient inference algorithm on trees and has demonstrated empirical success in numerous applications including low-density parity check codes; turbo codes and many more. Furthermore it is commonly used in pair wise Markov random fields, Bayesian networks and factor graphs.
For the purposes of explaining this algorithm, consider the marginal function, which is simply an unnormalized marginal distribution with a generic global function g(x).
Generalized Belief Propagation (GBP)
BP algorithms are presented as messages most of the time updating equations on factor graph. This involves messages between variable nodes and their neighbouring factor nodes and vice versa.
BP has its own difficulties for community detection and these are:

- The graph is highly connected with long-range links between all nodes. This means that there are too many variables to use BP for these long range links.
- You need a spontaneous symmetry breaking to get a group assignment

When extracting information from the BP algorithm, we assign each node to the most likely community for that node. This is used rather that the probable assignment of communities for all nodes. The other way is the idea of maximum accuracy. By this we are looking at a scenario if by chance a given node connects to more nodes of a different community then it will be classified wrongly so we could have a maximum of about 92 % for $Z_{in}= Z_{out}= 8$ with q = 3.
After talking about how to extract information from the BP algorithm, we now describe the implementation of the BP algorithm for community detection. This can be done in a number of ways.

- Solving the BP equations iteratively. The following should be chosen; pick a random edge and update belief then pick a random node and update belief. This can be done a number of times thereafter the belief should be replaced with a weighted average of old belief and solution of BP equations.
- Then there is the possible failure mode in which we look at the behaviour of BP equations and we realize that they don't converge. From diagnostics beliefs do oscillate. With this differences or diversions the solution is to have more iterations in order to have a marginal change on the iteration.
- Another thing in the failure mode is that BP equations do not break symmetry—Beliefs normally converge to a symmetric solution.

As we conclude our description of Belief Propagation we can say it is accurate.

Radicchi proposed a similar methodology with Girvan-Newman [43] but used a new metric –edge clustering coefficient whose computation time is less than Girvan-Newman's (GN) betweeness centrality which decreases Radicchi's time complexity to *O* (e^2) as compared to GN's O (e^3). In spite of the less time complexity in comparison to GN's algorithm, Girvan-Newman urges that the "edge betweeness algorithm" is computationally costly. This is shown in the evaluation of the score where all edges require a time order of MN, where M is the number of edges and N is the number of nodes. The iteration of the procedure for all N edges leads in the worst case to a total scaling of the computational time as M^2N which makes the analysis infeasible for moderately large network. For example if N is of the order of 20000, it is quite hard.

Given the weakness mentioned by Girvan-Newman about Radicchi's methodology, they came up with some ways of how to improve the algorithm. They start by saying that the algorithm that builds the tree just selects sub graphs that are candidate to be considered communities. In order for some one to check if they are actually such sub graphs, there needs to be a precise definition. IN the final analysis, if the sub graph does not meet the criterion, the sub graph isolated from the network is not a community a d the corresponding branch in the dendrogram should be drawn.

The precise definition that we talked about above can be that a community is generally thought as a part of a network where internal connections are denser than external ones. As a means of sharpening the use of detection algorithms, a more precise definition is needed. This is bent on the implementation in the algorithms of two plausible definitions of community which lead to the following formula.

The basic quantity to consider is k_i, the degree of a generic node *i*, which in terms of the adjacent matrix $A_{i,j}$ of the network G is $k_i = \Sigma_j A_{i,j}{}^2$. If we consider a sub graph $V \subset G$ to which node *i* belong, we can split the total degree in two contributions.

$$k_i (V) = k_i^{in} (V) + k_i^{out} (V)$$

$k_i^{in} (V) = \Sigma_{j\in V} A_{i,j}$ is the number of edges connecting node *i* to other nodes belonging to V.

$k_i^{out} (V) = \Sigma_{j \notin V} A_{i,j}$ is the number of connections towards nodes in the rest of the network.

The sub graph V is a community in a string sense if $k_i^{in} (V) > k_i^{out} (V)$, $\forall_i \in V$.

In a strong community each node has more connections within the community than with the rest of the graph. The sub graph V is a community in a weak sense if

$\Sigma_{i\in V} k_i^{in} (V) > \Sigma_{i\in V} k_i^{in} (V)$

In a weak community the sum of all degrees within V is larger than the sum of all degrees toward the rest of the network. We can say that a community in a strong sense is also a community in a weak sense but the reverse is not true.

In summary we can say that the GN algorithm goes through the following iterations. First choosing a definition of a community then computing the edge betweenness for all edges and removing those with the highest score. If the removal does not split the sub graph then you go back to the second iteration.

In the event that the removal splits the sub graph, test if at least two of the resulting sub graphs fulfill the definition. If they do then one can proceed to draw the corresponding dendrogram. So the procedure continues by iterating from the second step until all the edges in the network have been identified. We can say that Girvan-Newman algorithm is computationally expensive because it requires the repeated evaluation for each edge in systems, of a global quantity, the betweeenness whose value depends on the properties of the whole system. The result of completely analyzing a network turns out to grow fast with its size thus making the analysis infeasible for networks of the size larger than 15000 nodes [44].

As a remedy to this time problem, Radicchi et al came up with a divisive algorithm which requires the consideration of local quantities only. The elementary component of a divisive algorithm is a quantity which can single out edges connecting nodes belonging to different communities. As earlier mentioned this divisive algorithm employs edge clustering coefficient which is defined as the number of triangles to which a given edge belongs, divided by the number of triangles that might potentially include it, given the degrees of the adjacent nodes. The edge clustering coefficient can be expressed as

$$C_{i,j}^{(3)} = Z_{i,j}^{(3)} / m_{in}\,[(k_i\text{-}1)\,(k_j\text{-}1)]$$

Where $Zi_{,j}^{(3)}$ is the number of triangles built on the edge and $m_{in}[(k_i\text{-}1)\,(k_j\text{-}1)]$ is the maximal possible number of them.

The reason quantity is used in a divisive algorithm is that edges connecting nodes in different communities are included in few or no triangles and tend to have small values of $C_{i,j}^{(3)}$. Also there is the fact that many triangles exist within clusters. We can then say that $C_{i,j}^{(3)}$ is a measure of how inter-communitarian a link is. In the event that $C_{i,j}^{(3)} = 0$ meaning that the number of triangles is also zero, we consider a slightly modified quantity by using the number of triangles plus one in the numerator as shown below:

$$C_{i,j}^{(3)} = Z_{i,j}^{(3)} + 1 / m_{in}\,[(k_i\text{-}1)\,(k_j\text{-}1)]$$

If we take into consideration higher orders cycles we can define coefficients of order g in much the same way.

$$C_{i,j}^{(g)} = Z_{i,j}^{(g)} + 1 / S_{i,j}^{(g)}$$

Where $Z_{i,j}^{(g)}$ is the number of cyclic structures of order g the edge (i, j) belongs to while $S_{i,j}^{(g)}$ is the number of possible cyclic structures of order g that can be built given the degrees of the nodes.

After describing how some of the algorithms detect communities, we can say that the detection of the community structure is still an open challenge. Analyzing a network is requires that one specifies quantitatively and unambiguously what a community is provided that a definition is given it is in principle possible to determine all sub graphs of a given network that fulfill the definition. Realizing sub graphs is practically a hard thing to achieve even for small systems. So we can say that the search for the community structure has generally a more limited goal which is selecting, among all possible communities, a subset of them organized hierarchically, a denodrogram. This task is possible with the help or use of divisive and agglomerative algorithm. However, we must observe that comparison of such algorithms is non trivial.

1.3 Distributed Community Detection

Humans make up many communities and relationships. Two people can be strangers, familiar strangers, community members, friends or families. The two key criteria by which to categorize these relationships could be contact duration and number of contacts which are correlated to familiarity and regularity.

Pan Hui et al [46] introduced three distributed community detection algorithms which they called simple, *k*-clique and modularity. The common terminologies for all these algorithms are local community, adjacency set, boundary set, familiar set boundary-adjacency matrix and local modularity. We shall go o to define each one of the terminologies.

Local Community—A vertex's local community denoted by C, contains all the entries in its familiar set (Direct neighbours normally make up the familiar set) as well as the vertices that are selected by the algorithms that were advanced by Pan Hui et al. as we shall see later on.

Vertices in the same community may detect a different local community and this may be due to temporal non-synchronization. The structure of the algorithms that are mentioned earlier is as follows

A mobile device v_0 initializes its community detection procedure first, while the local community C_o only contains this source vertex. When this mobile device v_o comes across another device $v_{i,}$ they exchange part of their local knowledge of the network. The initial mobile device v_o then has to decide on the following based on certain acceptance criteria.

(a) Where to place the encountered vertex v_i because it has its familiar set F_o and local community C_o.
(b) The question of whether C_o should merge with the whole part of $C_{i.}$

All the algorithms introduced by Pan Hui et al differ only in the admission criteria into the familiar set, local community and merging of communities above.

Familiar set—The assumption is that each vertex will keep a map of vertices they have encountered with the corresponding cumulative contact durations. Vertices that exceed certain threshold values T_{th} are promoted to be included into its familiar set *F*. With reference to graph theory, we can say that these two vertices now have an undirected edge between them. Any given vertex v_i has a perfect knowledge of its own familiar set denoted by $F_{i.}$ Also any given vertex may have gathered some incomplete knowledge of other vertices familiar sets v_j and would be denoted by F_j.

It is imperative at this point that we give a brief introduction to Modularity before we continue with the other terminologies. Modularity is a variation of Clauset's community detection using local modularity [47]. Clauset goes on to define a measure of local community structure and an algorithm that deduces the hierarchy of communities that enclose a given vertex by exploring the graph one vertex at a time.

Adjacency set—The adjacency set of a particular local community C_o is denoted by u_o . From the graph theory point of view, It is the set of vertices which are outside the local community $C_{o,}$ but each vertex in it has direct edges connecting to one or more members of the local community $C_{o.}$

$$u_o = \{v_i \mid v_i \in \cup_{vj \in Co} F_j \setminus C_o\}$$

Boundary Set—For a given vertex v_o and its local community C_o , the associated boundary set B_o is defined as the subset of vertices in C_o , whose members have edges connecting to one or more vertices outside C_o .

$$B_o = \{v_i \mid (v_i \in C_o) \wedge (F_i \setminus C_o) \neq \varnothing \}$$

Boundary-Adjacency Matrix—For a vertex v_o and its boundary set B_o , the associated boundary – adjacency matrix of B_o is defined as $(B_o)_{ij} = 1$ if vertex v_o has knowledge tah the vertices v_i and v_j are connected, and at least one of them is in the boundary set B_o, otherwise $(B_o)_{ij} = 0$.

Therefore we can say $(B_o)_{ij} = 1$ iff

$$(v_i \in B_o \vee v_j \in B_o) \wedge (v_i \in F_j \; v_j \in F_i)$$

Local modularity—The entire local knowledge, G_o of a given vertex v_o that its has for the network, is made of the vertices in C_o and $u_{o,}$ together with its partial knowledge of the connections between those vertices. Clauset introduced local modularity with the view that each vertex can measure the sharpness of its local community boundary and the measures are independent of the size of the enclosed communities. The Local modularity R for a given C_o with B_o is given as

$$R_o = I \,/\, |T| \text{.............. (i)}$$

Where T is the set of edges G_o with one or more end points in B_o and I is the number of these edges with neither endpoint in u_o. If $Bo = \varnothing$, Ro is defined to have value of 1.

Addition of a new vertex $v_j \in u_o$ to an existing community C_o with B_o is possible and the change in the R value ΔR_o can be calculated as

$$\Delta R_o = x - R_o * y - z\,(1 - R_o)/\,|T| - z + y \text{ .... (ii)}$$

Where x is the number of edges in T that terminate at v_j
y is the number of edges that will be added to T by the agglomeration of v_j
z is the number of edges that will be removed from the T by the agglomeration.

Next we shall highlight the algorithms as advanced by Pan Hui et al. They use simple, *k*-clique and modularity to denote the variations of the skeleton algorithm.
When the mobile device v_o meets another mobile device v_i , the following algorithm will execute:
(a) The vertices v_o, must each maintain this information below.

- A list of encountered nodes and their contact durations.
- The familiar set F_o
- Its local community C_o detected so far
- A local approximation of the familiar sets of all vertices in its local community C_o.

$FS_oLC_o = \{F_j \mid v_j \in C_o\}$

(b) Initialization C_o is set to $\{v_o\}$, $F_o = \emptyset$ and FSoLCo $= \emptyset$
(c) At the point when v_o meets another v_i , they exchange local information which means that v_o will receive from v_i , the following C_i , F_i , FS_oLC_i . The first use of v_o 's newly acquired information is to improve its own approximation of F_i and FS_oLC_o . Merging of each local approximation of the familiar set in FS_oLC_o is done with the corresponding version in FS_oLC_i just obtain from v_i .
(d) v_o updates the total contact duration counter of v_i which is stored at v_o that's if v_i is not in F_o . This is done until v_i falls out of contact and meanwhile the algorithm forks and proceeds to the next stage. After a certain threshold v_o inserts v_i in F_o and C_o .
(e)In the even t that v_i is not in C_o, the v_i should be added to C_o if it satisfies the following algorithm-specific criteria
(i) For the simple algorithm
If $|F_i \cap C_o| / |F_i| > \lambda$ where λ in this case is the merging threshold.
(ii) For the *k*-Clique
If the familiar set F_i contains at least k-1 members of the local community,

$|F_i \cap C_o| \geq k\text{-}1$

(iii) For modularity
If $(F_i = \emptyset) \wedge (F_i \subseteq C_o \wedge B_o \neq \emptyset) \vee \Delta R_o > 0)$
(f) In the even that v_i is added to C_o in the previous step, the aggressive variants of the algorithm take the following behaviour.
For the simple
If the number of vertices overlapping C_o and C_i is higher than γ of $|C_o \cup C_i|$ then merge the two communities. γ is the merging threshold which could be different from λ.
So we have the merging criteria as

$|C_o \cap C_i| > \gamma |C_o \cup C_i|$

For the *k*-clique
Each vertex v_j is considered inside C_i if its familiar set F_j contains at least k-1 members of C_o , v_j is added to the local community C_o .
For modularity, the algorithms consider only adding the vertices in the set k.
From the above description of the algorithms, the simple algorithm requires less storage and less computation. The *k*-clique algorithm is in the middle and modularity is the most demanding one as it requires the re-evaluation of ΔR in each iteration. Therefore in the last step of the algorithm, only part of the community is considered to be merged, as a resource/performance tradeoff.

1.4 Genetic Algorithms

This sort of algorithm was first proposed in [48] as an optimization method in artificial intelligence. This algorithm is a nice method to use in a scenario where the solution space of a problem is very large and an exhaustive search for the exact solution is not possible. Potential solution members in the solution set need to be represented in a suitable data representation. Each member in a chromosome, which is the solution set represents a possible solution to the problem and the algorithm tries to find the best fitting solution member. As a means of improving the quality of the chromosomes, the algorithm uses genetic operations on possible solution members (chromosomes) for a given number of iterations. The chromosomes are initialized at the beginning and then for a given number of iterations, it uses a fitness function which assigns a fitness value to each chromosome and this shows how good a chromosome is to solve the problem. It goes on to reproduce chromosomes for the new population who will be used in the coming iteration. This is possible by performing cross over between members selected according to their fitness values.

The advantage of a genetic algorithm is its speed for converging a problem to a smaller solution space, and if it has a good fitness evaluation function, the solutions produced are near optimal. Also the algorithm's strong point is its crossover mechanism producing better solution members for the next generations. Mursel Tasgin et al [51] propose an algorithm in which they use a number of terminologies similar to other algorithms and these include network modularity, crossover, mutations, clean-up and community variance.

Network Modularity—the value in network modularity is used as the fitness value for each chromosome (solution member) and modification is done on the genetic algorithm that we discussed earlier. Mursel et al's algorithm starts with initial population creation. An integer array or vector *a* is used for data representation of community detection problem. The community identifier of the nodes is stored by the vector. So we have a_i as the community identifier of the node i . There are a number of chromosomes holding different community configuration information in the population. Initially each node is assigned a random community identifier. We need a mechanism to give a bias for initial placement of nodes into communities. Two nodes in the same community should have connectivity with each other.

Cross-over—this operation in the genetic algorithm is done by selecting two chromosomes according to their fitness values. Thereafter a cross-over point in the chromosome is selected. Transfer of the community identifiers of nodes in a community to nodes in the destination chromosome is done. The details of the cross-over in Mursel et al's algorithm is they name the chromosomes taking place in cross-over as source and destination. Then a random selection of a community is done from the source chromosome. Nodes are iteratively search that belong to the community and transfer the community identifiers of those nodes in source chromosome to nodes in destination chromosome.

Mutations—After the cross-over, the mutation is performed to some of the randomly selected chromosomes. A node is placed in a random community in the network when it is in mutation function.

Clean-up—this is as a means of improving the quality of the community splits that are needed to reduce the number of nodes that are placed in wrong communities. Although the overall fitness value is good for a community split, there may be a small number of misplaced nodes that does not affect the overall fitness value very much. The clean-up process is based on a new metric that is called community variance.

1.5 Shortcomings of the present algorithms

Most of the available algorithms are successful in community detection. But many of these algorithms have the issue of time complexities that make them inappropriate for large networks. More so some algorithms have data structures like matrices which are really hard to implement and use in the large networks. Then there is another issue that some algorithms have got to have past knowledge about the community structure. This is however the ideal scenario and not possible in real life networks. Then there are other algorithms that need threshold values and this becomes an issue because of the variant nature of different complex networks.

For the distributed community detection, the familiar set threshold values that were used in the emulations are trace dependent and are chosen based on the trace of the whole duration. In the present values of the algorithms, we need to specify a static familiar set threshold but probably in future more dynamic methods will be in place to reduce on the manual configuration.

2.0 Properties of Community detection algorithms

In social network analysis a group of algorithms focus on the discovery of the cohesive sub-structures which include cliques, n-cliques, n-clans, n-plexes as well as the quasi cliques. The dense sun structures often impose extra restrictions on the community definitions. The definition of n-clique requires that the distance between any pair of vertices should be no more than n and for quasi clique the ratio of the number of each vertex's neighbours to the number of all the vertices in the sub structure is not less than a threshold value. The other widely used technique in social networks analysis is the hierarchical clustering which groups similar vertices into larger communities.

For the properties of a modular network, we shall consider a network made of m identical complete graphs that are not joined together. The modularity in this case is maximal for the partition of the network into the cliques and is given by the sum of m equal terms. Each graph has l=L/m links with a total degree of d=2l.

The modularity of a partition of a network [6] can be written as

$$Q = \sum_{s=1}^{m} \left[\frac{ls}{L} - \left(\frac{ds}{2L} \right)^2 \right] \text{..................} \quad \text{(i)}$$

Where the sum is over the m modules of the partition, *ls* is the number of links inside modules, L is the total number of links in the network and *ds* is the total degree of the nodes in module S. The first summand in equation (i) is the fraction of links inside module S. The comparison with the null model leads to the quantitative definition of community embedded in equation (i) above.

The conclusion is that a sub graph S with *ls* internal links and total degree *ds* is a module if

$$\frac{ls}{L} - \left(\frac{ds}{2L} \right)^2 > 0 \text{ ....} \quad \text{(ii)}$$

We can represent the number of links ls^{out} joining nodes of the module S to the rest of the network in terms of *ls*.

So we have $ls^{\text{out}} = a\, ls$ with $a \geq 0$

Therefore $ds = 2ls + ls^{\text{out}} = (a + 2)\, ls$ and so the equation (ii) becomes

$$\frac{ls}{L} - \left[\frac{(a+2)}{2L} ls \right]^2 > 0 \text{ ....} \quad \text{(iii)}$$

On rearrangement we have

$$ls < \frac{4L}{(a+2)^2} \text{ ....} \quad \text{(iv)}$$

The sub graph S is a disconnected part of the network and is a module if *ls* < L and a=0; and normally this is true. If a remains positive the above equation (iv) sets up an upper limit to the number of internal links that S must have in order to qualify as a module. The above equation means that the modularity is dependant on the size of the entire network other than dealing with a local comparison between the number of internal and external links of the module. The attributes "Internal" and "external" mean that the degree is calculated with due consideration of the internal and external links. That is for a < 2 one has $2ls > ls^{\text{out}}$ implying that the total internal degree of the sub graph is larger than the total external degree.

$$ds^{\text{in}} > ds^{\text{out}}$$

Given a <2, the value $\frac{4L}{(a+2)^2}$ is in the interval $\left[\frac{L}{4}, L\right]$. This implies that a sub graph of size *ls* such that a<2 and *ls* is less than a quantity in the interval $\left[\frac{L}{4}, L\right]$ would be taken as a community both in the context of modularity and the definition of Radicchi *et al.*

Below we have the sufficient conditions for which these constraints are met normally

$$ls < \frac{L}{4} \text{ and } a < 2 \text{ ………………… (v)}$$

From equation (ii) a partition of a network into actual modules would have positive modularity; since all summands in equations (ii) are positive.

A value of Q larger than 0.4, normally indicates that the sub graphs of the corresponding partition are modules. The maximal modularity differs from one network to another and depends on the number of links of the network. Finally from the above analysis we derive the maximal possible value $Qm(L)$ that Q can attain on a network with L links.

Since there are not links connecting nodes of the graph to other graphs. So we finally have

$$Q = m\left[\frac{l}{L} - \left(\frac{2l}{2L}\right)^2\right] = m\left(\frac{1}{m} - \frac{1}{m^2}\right) = 1 - \frac{1}{m} \text{ ……………………… (vi)}$$

In the limit as the number of graphs tends to infinity the modularity Q converges to one. In equation (i) we realise that the result is still valid even if the m connected components are not graphs. The other point is that the number of nodes of the network and within the modules does not affect modularity. For topological constraints such as connectedness, we need to have L/m links inside the modules for m modules.

We also look at a scenario where we can construct a connected network with n nodes and L links with maximum modularity. In order to achieve it, we shall go through two steps. Firstly we look at the maximal value $Qm(m, L)$for a partition with a fixed number of modules m; after that, we look for the number m* that maximises$Qm(m, L)$.

To maximise the contribution to modularity of each module, we endeavour to reduce the number of links connecting modules as much as possible. All this is done considering a partition of m modules. To observe connection in the network, we must have at least m-1 inter community links.

To keep it simple we shall analyse the simple ring like configuration illustrated in figure 1 which has m inter community links instead of m-1.

The modularity of such a network is

$$Q = \sum_{s=1}^{m}\left[\frac{ls}{L} - \left(\frac{2ls+2}{2L}\right)^2\right] \text{ …………………… (vii)}$$

The equation (vii) reaches its maximum when all modules contain the same number of links i.e $ls = l = \frac{L}{m} - 1$ ∀s = 1,2 ,3 ….., m

The maximum is then given by

$$Qm(m,L) = m\left[\frac{L/m-1}{L} - \left(\frac{L/m}{L}\right)^2\right] = 1 - \frac{m}{L} - \frac{1}{m} \quad \text{…………………….. (viii)}$$

In order to find the maximum of $Qm(m,L)$ leaving the number of modules m variable, we take the derivative of $Qm(m,L)$with respect to m.

$$\frac{dQm}{dm}(m,L) = -\frac{1}{L} + \frac{1}{m^2}$$

Which reduces when $m = m^* = \sqrt{L}$ and it corresponds to the absolute maximum $Qm(L)$ of the function $Qm(m,L)$[45]. This proof is completely general and does not require preliminary assumptions on the type of network and modules. We therefore stick to the real-valued expressions with the meaning more clear. So we have maximal modularity as

$$Qm(L) = Qm\left(m^*, L = 1 - \frac{2}{\sqrt{L}}\right)$$

In limit as L → ∞ $Qm = 1$

For each module $1 = \sqrt{L-1}$ has the corresponding number of links. The idea that all modules have the same number of links does not mean that they do have the same number of nodes. Furthermore modularity does not depend on the distribution of the nodes among the modules as long as the topological constraints are satisfied.

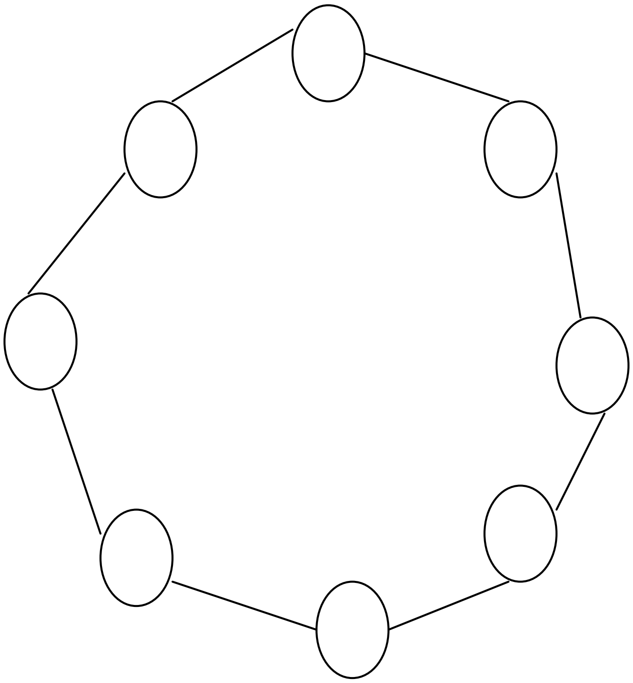

Figure 5 Design of a connected network with maximal modularity

Next we shall talk about the resolution limit by analyzing a network with L links and with up to three modules which satisfy the conditions $ls\langle\frac{L}{4}$ and a < Z as shown in figure 6. We shall base our discussion on pair of modules, M_1 and M_2 and distinguish three types of links. First those links internal to each of the two communities (l_1 & l_2 respectively) then the link between M_1 and M_2 (l_{int}) and lastly the link between the two communities and the rest of the network M_0 (l_1^{out} & l_2^{out})

For simplicity, we express the number of external links in terms of l_1 & l_2 implies that

$l_{int} = a_1l_1 = a_2l_2$

$l_1^{out} = b_1l_1$ and $l_2^{out} = b_2l_2$ with $a_1, a_2, b_1, b_2 \geq 0$

We also have $a_1 + b_1 \leq 2$, $a_2 + b_2 \leq 2$ and $l_1, l_2 < L/4$ and this is so because M_1 and M_2 are modules by construction.

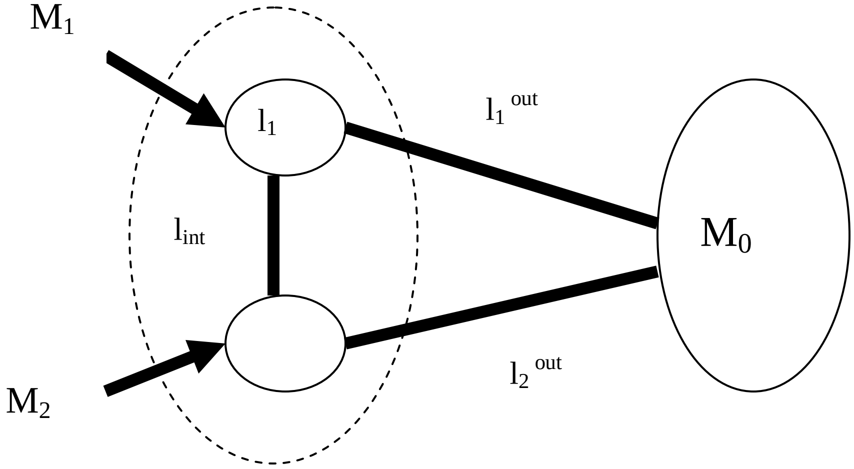

Figure 6 A three module network with two modules M_1 and M_2 and M_0 as the general network

The networks that we have examined are relatively small but the problem that seems to linger on is the idea of increasing on the network size—this is true when small communities coexist with large ones and the module size distribution is broad. As an example we can look at the Amazon the online seller. What happens is the Amazon recommends items that have been bought by people who have bought the same product. This way it helps build a network in which nodes are the items for instance music or books, and there is an edge between two items A and B if B was frequently bought by buyers A.

In this section, we have analysed in detail modularity and its applicability to community implied by modularity is actually not consistent with its optimization, which may favour network partitions with groups of modules combined into larger communities. One suggestion that comes up is that by enforcing modularity optimization, the possible partitions of the system are explored at coarse level—meaning that modules smaller than some scale may not be resolved. The resolution limit of modularity does not depend on particular network structures, but results only from the comparison between the number of links of the interconnected communities and the total number of links of the network.

From what has been discussed, it is not possible to rule out that modules of virtually any size may be clusters of modules, although the problem is most likely to occur for modules with a number of internal links of the order $\sqrt{2L}$ or smaller. Therefore it is vital to check the structure of all detected modules, for example by constraining modularity optimization on each single module, a procedure which is not safe but which might give useful indications. The basis of the resolution scale lies in the idea that modularity is a sum of terms where each term corresponds to a module. So we can say that the maximal modularity is the equivalent of looking for the ideal trade-off between the number of terms in the sum which is literally the number of modules and the value of each term. What we observe here is than an increase in the number of modules does not necessarily correspond to an increase in modularity because the modules would be smaller and so each term of the sum would be smaller. This may explain why for some characteristic number of terms, modularity has a peak. The problem is due to the optimal partition imposed by mathematics does not necessarily capture the actual community structure of the network, where communities may be very diverse in size, that is circumstances where the network is large.

Other quality functions other than the modularity may have intrinsic resolution scale that undermines their reliability. But the assumption is that quality functions are mathematically similar to modularity. This not withstanding there are many possible ways to define the quality of a partition; for example one could take the average quality of the modules, instead of the sum and obtain very different results.

The idea that quality functions such as modularity can have an intrinsic resolution limit gives rise to a new theoretical framework that focuses on a local definition of a community other than on a definitions based on a global null model. Lastly we can say that quality functions are still helpful but their role should probably be limited to the comparison of partitions with the same number of modules.

3.0 Dominance detection in animal packs

When it comes to the biology we shall look at animal behaviour. Animal behaviour relates to what an animal does and why it does it. The animal behaviours are genetically determined or instinctive and other behaviours are learnt. Following we look at the behaviour in a little more detail.

3.1 Instinctive behaviour or Genetic behaviour

Genetic behaviour is very much based on a fixed action pattern according to the people who study behaviour. Honey-comb making in bees is a good example of genetically determined behaviour. There is little or no variation in the way bees make honey-combs. Many of the bees whether the so called tamed or wild build there honey-combs the same way. For this genetically determined behaviour fixed action patterns do not need learning they just come naturally.

Identification of fixed action patterns cannot be done by generalizing because they are highly stereotypic and species-specific. We can for instance say that the behaviour of dogs based on their up bringing may fit the instinctive behaviour but is precisely learnt behaviour. Dogs that just grow up in the absence of human beings or training are deprived of the experience of learning to live with people—and dogs that grow up in the wild have a totally different behaviour from the ones that get training. IT is possible to distinguish a behaviour that is a fixed action pattern from those which are learnt. This is achieved by carrying out deprivation experiments in which animals are raised without parents, or contact with their own species. During some of these deprivation studies, there may be lack of expression of a particular behaviour but this does mean that it is not genetically determined. It may only indicate that the needed stimuli to elicit the expression of fixed action patterns are called releasers. Another state in which a fixed action pattern may not be displayed is when an animal is not in the appropriate physiological or developmental state.

3.1.1 Learnt behaviour

We shall start by defining what Learning is and it is the modification of behaviour in response to specific experiences. This learnt behaviour is categorized in many ways and these are Insight Learning, Observational learning or modeling and Associative Learning. We shall discuss them in detail.

3.1.2 Insight Learning

This is considered the highest form of learning observed. It is the ability to solve a problem or to execute appropriate behaviour the first time the animal is exposed to a situation. For instance a trained dog can be able to open a slightly closed door by using one of its front legs without necessarily having gone through that training. This learning is not limited to only a specific group of animals but many are able to exhibit this kind of action.

3.1.3 Observational learning or modeling

This normally comes out of observation done by an animal while observing other animals. A classic example here could be in a pack animal such as the wolf, whose hunting behaviours fit this category. We shall talk more about Wolves as our case study.

3.1.4 Associative Learning

In this sort of learning, an animal learns to associate one stimulus with another. Still this learning can take on two forms and these are classical conditioning and operant conditioning. Classical conditioning was investigated by Pavlov in his well known dog experiments. He went on to stimulate the dogs to salivate by rubbing meat powder on their lips. The meat powder scent stimulated salivation as a physiological process. From there he conditioned the dogs by sounding a tuning fork at the same time applying the meat powder. This revealed that the animals had been conditioned to associate the sound of the bell with the meat powder and would salivate to the noise without the presence of food.

The other form is the operant conditioning and in this an animal conducts a chance action. That is the animal performs or carries out a specific action then it is rewarded with food. If this goes on for a while the animal realizes that if it performs that particular action it is rewarded with food and subsequently will keep carrying out that action because it expects food. This kind of learning is many used in most animal learning.

However, in the event that the reward for this action is not given, the animal ceases to carry out the appropriate response to a stimulus.

The debate between human psychologists as well as animal behaviourists focuses on the relative importance of instinctive and learned behaviours. The argument here is that the environment in which the animal is placed determines the learned behaviour. This debate we are talking about is referred to as Nature versus Nurture. Nature is about genes and Nurture about the environment. Balancing between fixed and learned behaviour varies with species. A large part of our (human) behaviours are learnt. Many humans learn some of their behaviours from role models and without role models to learn from there is a possibility of great emphasis on programmed behaviour. Certain behaviours have to be fixed as a matter of survival as there may never be a second chance to learn them. An example here is the kangaroo rat instinctively reacts to the sound of a rattlesnake with an escape jump.

There are critical periods for the development of the appropriate behaviour. A classic example is a dog which may be unable to socialize well with humans if it comes to live with humans after 14 weeks.

The imprinting process must occur in a distinct, usually short time. It is normally irreversible and involves an attachment to an object that will evoke subsequent adult behaviours.

3.2 Communication among animals

This relies upon the abilities to perceive sensory information. It could be visual, auditory and olfactory (smell). Due to the fact that domestic animals perceive the world in a different way to humans, they respond differently as well.

Different animals have different ranges of monocular vision. The vision range in animals depends much on the placement of the eyes. For example a cat has a much smaller arc of vision as compared to a horse because of the eye placement. But more than half of this is binocular vision.

Overall all domestic animals have some ability to discriminate colours and most have better night vision to humans. There are different ranges of hearing for auditory sensing as well as acuities. Sheep and dogs can discriminate higher frequencies than humans. On the contrary cats do have almost the same range of hearing like humans.

The most important sense of domestic animals in terms of communication is the Olfactory (smell) sensing [10]. In this category dogs seem to have the greatest olfactory ability of the domestic species. They have the ability to sense many compounds a 1/100 the concentration of humans and also for many weeks after they have been placed. The communication methods employed by animals in response to stimuli perceived through these sense organs are the same types as we outlined earlier. However these communication methods have varying importance between domestic species.

Pigs probably have the most complex set of domestic animal vocal sounds. Horses, cats and dogs follow in the queue. These sounds have various meanings that range from greetings, excitement, aggression, fear, pain etcetera.. As we stated earlier horses have a wider vision range but the ear position can tell a lot about the animal’s disposition. In as much as sounds indicate some thing also the visual things imply certain things. In horses for instance the ears pointing backwards generally imply aggression and the flatter the ears to the head the greater the aggression. In the cat, a high tail is a greeting or sign of being curious. Pigs and horses are known to show grooming behaviour as do many of the monkeys and primates. The other funny observation is that the subordinate pigs groom dominant ones while in horses those with comparable rank groom each other [49].

Olfactory or smell clues and scents are used as a way of marking territory and also showing the way home. These clues also distinguish individuals say for instance in cattle olfactory stimuli play a major role in heat detection.

3.3 Behaviour and detection of animal disease

Trying to understand domestic animal behaviour is not only interesting but is also crucial in their management, productivity and welfare [50]. Many times experienced vets or animals owners can identify that there is a problem especially from listening to the sounds made by the animal or by observing the change in posture. Some sicknesses of problems can be easily detected while others are quite hard to detect. For instance if a horse is seen sweating then it is probably a sign of acute pain and if a dog is seen tilting its head then the problem is probably with the ears. Then for cows, if the order in which the animals comes to the milking parlour changes, especially for a dominant animal then there is a problem.

3.3.1 Breeding and Feeding Control

We can say that a domestic animal is one which has been transformed by selective breeding and control of the food supply by humans. Understanding the behaviour of animals helps in identification of feeding problems as well as being able to modify maternal behaviour to accept orphaned animals. It also helps in the manipulation of group numbers and the size of animals thereby reducing on aggression [50].

3.3.2 Environment and Behaviour

A lot of information can be got out of observing animal behaviour and the way the animals cope in a given environment. This can lead to better design in housing and in the management systems for the farm animals. A classic example could be climate which includes changes in temperatures which could be very cold or very hot. This will definitely lead to behavioural change which can be so easily identified. Also looking at it from another angle cows may end up just standing in stalls instead of feeding and this is an indication that the stalls are uncomfortable [50].

3.3.3 Training

Positive reinforcement of desired behaviours also referred to as associative conditioning is one of the ways that is relied on most when training domestic animals. However, the genetic design is the most important. The genetic predisposition of the border collie to herd is adapted by having regular training in the bid to increase skills and control that instinctive ability. It is a fundamental aspect of animal science with respect to the comparison and recreation species.

3.4 Case Study Wolf Packs (Alpha Pair)

It has been observed that Wolves are loyal to their mates. Like in humans, at one point some wolves that get tired of being ordered around by their parents and so they leave with the hope that they will start their own pack. They end up finding another wolf and with cooperation they can start up another family. These wolves are normally referred to as the alpha pair. There exists a bond between the alpha pair that is not seen among the rest of the pack.

Wolves were at one point thought of as being monogamous and the assumption is that they mate for life so that if one of the couple dies, the other does not seek another mate. But this is not true as wolves have been known to have many mates throughout their life time. In fact in certain circumstances mates leave their mates in search of new ones. In some of the rare occurrences, there have been two breeding females to one male in a single wolf pack.

The alpha pair is extremely loving towards each other, spending much of the time together. When it comes to the breeding season, the wolf pair get more affectionate with each other. The assumption is that the wolf society treats females as equal to the mates. All members of the pack both male and female hunt, play and eat together. Pregnant wolves as well as the females with cubs normally have preferential treatment in comparison to the rest of the pack. A lot of protection is given in addition to plenty of food. Younger members in the pack take turns looking after the cubs especially at the times when the older wolves have gone hunting.

3.4.1 Alpha Roles

Often times wolves maintain a hierarchy but when a question is posed—of the two the alpha male and alpha female, who is the most dominant many would give the answer that it is the alpha male. But what is true actually is that in most packs the two are equal. They normally have equal but different duties. For instance the alpha male's role is to scent mark, keep intruders at bay and making sure that the other males in the pack do not mate. At times the alpha males lead the hunt although many times the alpha females have been seen as leading the hunts. On the hand the alpha female's duties are to manage the females in the pack by prohibiting them from mating, choosing her mate and rearing the cubs. Her most important duty is choosing the den site which determines the life or death of the future generation.

Overall, the responsibilities of maintaining a wolf pack are shared equally with each one directing certain aspects. Generally female responsibilities are viewed as more important because she takes care of the tasks that create the pack. As a comparison human society is now bent on moving towards the same level that wolf societies have been for a long time that is equal opportunity leadership.

3.4.2 Community Naming Convention

Once all possible communities have been successfully detected a question may arise and that is what the discovered communities might be. Many times a community covers some kind of common relationship shared among the involved entities. Is it possible to find out what relationship exists? We shall use a general method to describe a specific community by combining the topology information and the natural attributes of the contained entities. The entire process results into creation of a profile for the community in mention.

Assume two data sets, one is the entity relational data which is modeled by graph G={V,E} with V being the entity (vertices) set and E the relation(edges) se respectively. Then we also have the entity attribute data R={a_0, a_1,….a_{n-1}, r_c}where a_0, a_1,….a_{n-1} are the natural attributes and r_c is the community by the previous process. With the basis of G and R the challenge is to find some key attribute values to characterize the given community.

There are two challenges for the naming tasks that is data gathering and synthesis of G and R in real applications which is rather complicated. You will find that many a times there are not enough attributes for the entities. The other issue is the classic approach to give a profile for a group of entities known as rule induction. But the problem is after discovery the communities; we are unable to use this method. The naming approach involves finding the central entities of each community and making use of the number of naming mechanisms based on the number of the attribute values held by the central entities.

Central entity resolution

Taking the self-similarity [53] and the scale-free properties into account, the assumption we make is that most sub graphs extracted from the social networks often have central entities in the same way like the whole social network often has several hub nodes. Central entities usually have a big impact on the overall formation and development of the given community. For instance in a collaboration network, the central entity of research team could advisor who originally started the research area, while in telecom networks , the central person in a group of frequently contacted people may be the chief officer in a department. In much the same way Wolves also have particular Wolves that are frequently contacted. The Wolves that are frequently contacted have alpha roles and therefore derive their name which is the alpha pair. Therefore the characteristics of the core figures greatly influence the properties of the community. There are many methods that have been proposed to quantify the centrality of an entity based on the topology of a given graph, such as degree, relative centrality, betweenness and PageRank. These approaches can bring us a unique value U(v) for each vertex v. On the overall, the greater the value is, the more important the vertex will be in the community. How to find out the central vertices according to these values is what we actually need to do. If the value of some vertex is significantly greater than those of the other vertices, or if there exists a large gap between two neighbouring vertices, such as w and v with U(v) > U(w) after ranking all of them according to the centrality measurements in the descending order, the vertices, whose value is more greater than that of v, are the central entities of the community. Below we describe the procedure in the algorithm.

Algorithm Central Entity Resolution (C, p)

```
a: {C is the given community and p is a threshold value}
b: calculate the centrality of each vertex $v_i$ in c
c: $v_0, v_1, ....... v_{n-1}$ is arranged by the descending order of their centrality $c_0, c_1, .......c_{n-1}$
d: i ⇐ 0
e: while i < n-1 do
f: if $\frac{ci - ci\text{-}1}{ci - ci\text{-}1}$ > p then
g: return {$c_k$ | 0 ≤ k ≤ i}
```

```
h: else $i \Leftarrow i+1$
i: end if
j: end while
```

4.0 Conclusion

Through out our study we have looked at and described the present community detection algorithms and we were able to give some of the general limitations of these algorithms. The most prominent one was the one of time complexity. Some other algorithms generally called the distributed community detection algorithms were also discussed and they seem to have different levels of computational complexities and resource requirements. These algorithms depended on human mobility datasets. After looking at the algorithms we would like to propose that in future evaluation of the available algorithms be done on more mobility traces such as WiFi traces.

Apart from the other algorithms we can say that the genetic algorithm tries to optimize network modularity using genetic algorithm methods. The algorithm is fats and scalable to very large networks due to its O (*e*) time- complexity. In this case *e* is the number of edges in the network. Furthermore this algorithm does not need priori knowledge about the community structure like say for example the number of communities in the network or the threshold values. This algorithm produces directly the community structure of the network in its results without any dendograms. Besides there will be no further processing needed.

Looking at one of our sub question as to whether examples from biology can be helpful in determining dominance in animal packs, we can say that we realize that the way detection of animal behaviour is done is totally different from the way the community detection algorithm work. Many times animals learn their behaviour out of instinct and from the environment in which they grow up.

So when it comes to community detection algorithms in our view biological examples cannot be helpful as the way in which the two operate is quite widely apart. However in some aspects there could work in much the same way and an area in mention is the area of priori knowledge. Some animals like we saw use the olfactory sense in trying to find their way home or trying to establish their territory. So this could be a similarity but this is one in the so many aspects that the community detection algorithms use in there quest to connect nodes and edges.

We can also say that we were not able to establish whether there are community leaders or not. In many of the algorithms they just come up with the different edges and nodes that seem to much the community structure that they are looking for.

All in all there is much that has to be done in community detection algorithms to make then faster that they are now as well as reducing on the complexities involved. The detection of the community structure is complex networks is a promising area of research with its own challenges. It involves being able to analyze a network and specify unambiguously the definition of a community. If a definition is in place it is in principle possible to determine all sub graphs of a given network that fulfill the definition. The search for the community structure has generally a more limited goal and that is selecting among many possible communities a subset of them organized hierarchically. There are the divisive and agglomerative algorithms that carry out these tasks. Comparing the performances of these algorithms is a non trivial task. In some simple case, the artificial graph with four subsets, it is possible to assess quantitatively the validity of the results. In the network of scientific collaborations, no quantitative measure is in place to decide, with a definite definition of a community.

5. Introduction to Part Two

RFID is an acronym that stands for Radio Frequency IDentification. It refers to small electronic devices that consist of small chip and an antenna. The RFID tag serves the same purpose as a bar code or a magnetic strip on the back of an ATM (bank) card. It provides unique identification for the object on which it is attached. In much the same as a bar code is scanned it is also scanned in order to get identifying information from it. One thing that is significant about RFID tags is that they do not need to be positioned precisely relative to the scanner. Normally at points-of-sale (POS) there could be problems making sure that a barcode is read and when it comes to the ATM card, it has to be swiped through a special reader. In contrast, RFID tags work within a few feet of the scanner (reader). For instance a customer may just have to set the shopping basket of items on the reader which queries the tags and totals up the purchase immediately.

Radio Frequency IDentification (RFID) has been in existence for a while now and most records available indicate that it came into force round about the World War II after the Germans employed it in differentiating enemy planes from their planes. At that time it was called Identify Friend from Foe (IFF) and since then it has seen a number of transformations and is still under going transformation as a lot of research is being done to realise all the potential there is in RFID.

RFID technology was adopted in use for identification of animals in the late 70s and is still being used up to now for the same purpose. There used to be a problem in identifying which animal had been treated for instance in preparation for travel and so it was imperative that a method of identification be devised. So RFID tags were put on each animal. RFID tags have the advantage that each tag has a unique identification number so there can be no problems to identify a particular item as compared to the bar codes (Universal Product Code (UPC)) which are the same for the same product worldwide. For example a 0.50 L bottle of Coca-Cola in Europe will have the same bar code as the same bottle in America.

They have been adopted in ubiquitous computing by some big organisations as a means of tracking goods in transit so as to be able to tell they have arrived at the destination or not among many other areas. Also in the supply chain there is the advantage that the supplier is able to realise the reduction in a certain product from a given supermarket and then begin to plan on the restocking process with out having to hear from the supermarket managers.

These are some of the merits of RFID technology but all this comes with a price. Having RFID technology work in such a way means that each item should be marked with a RFID tag—this is costly and cannot be sustainable at the moment. In our study we have looked at some of the available RFID authentication protocols and describe how they work and also mentioned the security properties they achieve.

6.1 RFID background

There are some claims in technology papers that the technology used in RFID was available as early as the 1920s and then also that RFID systems have been existent since the 1960s. However we shall rely more on the history of RFID right from the time of the Second World War when it was used for identification of enemy planes.

Before the IFF technology came into use radar had been invented in 1935 by a British physicist Robert Alexander Watson-Watt [1] and was in use by the Japanese, Germans, British and Americans. It was possible to help detect any planes that were in range but the problem was that it was not possible to differentiate between the planes coming back to base and those of the enemy. However, the Germans came up with some idea of having their planes rotate as they returned to base. They had discovered that this would alter the radio signal that was sent back to the control tower and as a result the Germans would differentiate their planes from the allied planes. The idea of IFF was in essence the first RFID system.

The British under leadership of Watson-Watt engaged in a project that sought to improve on the original radar so that they could identify their planes from the others. So they came up with the first IFF system and installed a transmitter on each British plane. This system worked in such a way that when a plane received signals from radar stations on the ground, it sent back a signal that identified it as a friendly plane. Precisely a signal is sent to a transponder which responds by reflecting a signal or by broadcasting a signal.

After this development there were more advances in the radar and Radio Frequency (RF) communication systems and in the period between the 1950s and 1960s a lot of research was done by various scientists and academicians in Europe, Japan and the United States [1]. Their notion was mainly based on the way Radio Frequency energy would be used in the identification of objects at a distance or objects that are far apart.

With these developments and innovations in place, companies came up with anti-theft systems which would help to determine whether an item was paid for or not and as they became popular the companies started selling them as a means of making some money. Lately or nowadays this development is used in electronic article surveillance tags used in packaging and the tags are 1-bit. This bit is normally on within the shopping mall area that is before the items goes through the readers and after the item has been paid for by the buyer, the bit is turned off normally by issuing a kill command at the counter of the shopping mall. The buyer is able to leave the shopping mall without any problem but if the item is not paid for, the bit remains turned on and if the item goes past the readers then the item will be detected and an alarm will go off.

The full implementation of RFID technology was evident in the 80s and different parts of the world developed it according to the needs they wanted solved [4, 5]. For instance in Europe the interest was in the areas of short range detection systems for animals especially when it came to treatment prior to transportation, then the other area was in industrial and business applications as well as in electronic tolling. On the other hand, the United States greater emphasis was on transportation and personnel access.

The use of RFID has since been used in a number of other areas like in the supply chain management where suppliers can monitor the shelves of the different stores they supply with various items and so are able to deliver the necessary quantities at the right time. Furthermore, companies are able to track their orders right from the manufacturing companies up to the point when they arrive at their premises or vice versa.

The first RFID patent for an active RFID tag with rewritable memory is believed to have been got by Mario W. Cardullo in January 1973 [2]. Also in the same year Charles Walton an entrepreneur received a patent for a passive transponder used to unlock a door without a key. About this time the U.S government through its energy department got involved in the research in RFID systems. It collaborated with Los Alamos National Laboratory to develop a system of tracking nuclear materials. The concept that was developed had a transponder in the truck and readers at the entry or exit points of secure facilities. Improvement of this kind of system led to the development of the toll payment systems which are being employed on roads, bridges and tunnels around the world.

RFID commercial systems starting operating at a frequency of 125 kHz and then later moved up to a higher frequency of 13.56 MHz which was unregulated and unused in most parts of the world. This offered the advantages of greater range and faster data transfer rates. The first systems to work on the higher frequency were used to track reusable containers in Europe but currently they are used for access control payment systems, contact-less smart cards and anti-theft devices in cars but to mention a few.

In the beginning of the 1990 decade, an ultra-high frequency (UHF) RFID system was patented by IBM engineers. It did offer longer read range of up to five metres and faster data transfer rates. Towards the end of the decade, the Auto-ID Center [3] was set up at the Massachusetts Institute of Technology (MIT) under the consortium of the Uniform Code Council, EAN International, Procter and Gamble and Gillette. This was a big boost to the UHF RFID systems.

Research was done by two professors David Brock and Sanjay Sarma and this was looking at the possibilities of putting low-cost RFID tags on all products to enable them to be tracked in the supply chain [7]. Before the research was done people thought of tags as a mobile database that carried information about the product. Sarma and Brock turned RFID into a networking technology by linking objects to the internet through the tag. This was a turning point in some businesses as a manufacturer could automatically let a business partner know when a shipment was leaving the dock at a manufacturing facility and likewise the retailer would automatically let the manufacturer know when the goods arrived.

In the first four years of the existent of Auto-ID Center [6], it opened up laboratories in Australia, United Kingdom, Switzerland, Japan and China after getting support from more than 100 large end-user companies along side the U.S Department of Defence. It developed the Electronic Product Code (EPC) numbering scheme and the network architecture for looking up data associated with an RFID tag on the internet. This technology was licensed to the Uniform Code Council in 2003 which lead to the creation of EPCglobal as a joint venture with EAN international. This marked the end of Auto-ID Center which and the rest of its work was passed on to Auto-ID labs.

6.2 RFID Technology

This is basically the technology used in the RFID systems. The RFID system is made up of three components namely the tag, reader and the supporting infrastructure normally comprising of both software and hardware.

Readers are devices that communicate with the tags by broadcasting radio signals. In the normal operation, the tag is expected to send information back to the reader. The readers can either be portable handheld devices or fixed terminals which are positioned in strategic places such as loading platforms, entrances and exits of buildings.

Tags also known as transponders are tiny devices made up of a microchip which has memory storage and an encapsulating material and an antenna. The tags are categorized in two ways, the read-only (passive) and read-write tags (active). The read-only tags have an identification code specifically referred to as the Electronic Product Code (EPC) and this is given and recorded at the time of manufacture or at the time of tagging an item. The data written on the tag in this category of tags cannot be altered or added on to but it can be read several times. The read-write tags have the provision of changing their memory and therefore creating the opportunity of writing data to the tag a number of times. They have the advantage that their identity codes can be changed and also that they offer much more functionality but at higher cost.

These tags can be read through a hard material or packaging material and also do not need line of sight as opposed to the Universal Product Code (UPC) commonly known as bar codes. These tags can be used to uniquely identify a single item from a collection of identical items to which they are attached. This is one of the characteristics of RFID that has led to privacy issues and concerns in the use of this RFID technology.

6.2.1 Chipless and Chip RFID

RFID tags can be classified as having a microchip or not. “Chip” tags contain an integrated circuit chip and the chipless ones do not. So this makes the latter cheaper and it may store up to 24 bits of information which is good enough for a company’s internal use, like in a warehouse. For mass market applications the chipless tags can not work as the memory is too low and they do not have the ability to read multiple tags within range.

Chip tag RFID enable data, such as a serial number or product code, to be stored and transmitted by portable tags to readers that processes the data basing on the needs of a given application. Chips currently in use are able to store 96 bits of data and this is good enough to include a manufacturer’s name, a product name and a number of unique numbers that could be assigned to products.

Finally we have the RFID specific software that translates the raw data from the tag into information about the items and any other details that are represented by the tags. The information is sent to other application for further processing. For the case of read-write tags software is such a vital part of the system as it controls whether the data can be written to the tag, and also locating the right tag to contain particular information as well as starting the process of affixing data to or altering data in the tag memory.

RFID technology has a lot of potential and a wide range of areas that it can be applied since it has the ability to have tagged items communicate when queried by readers. Some of the areas in which RFID technology is being used include the monitoring and controlling of the flow of goods during the manufacturing process and also from the factory to the retail stores and this is simply Supply Chain Management then it is also working in the controlling and monitoring of the movement of baggage from check-in to loading on the airplane and in patient care and management by providing a means to rapid and accurate verification of information concerning patient medical history.

6.3 RFID Privacy and Cost

The concept of privacy desires to be carefully studied. We cannot say that we have a specific definition for privacy because there are many proposals that have been submitted without a clear articulation of what privacy is. One thing to note here is privacy is a personal, subjective condition as it will vary from one person to another. Our take therefore is one way of protecting privacy is to distribute decisions about how personal information is used to the people affected.

In RFID systems, if privacy protection is not given the due attention then the users of the system will also susceptible to being followed and located wherever they are thus encroaching on their privacy. An RFID system can easily protect users' privacy by making use of the public key cryptography but it is viable in low cost RFID tags which are getting adopted in ubiquitous computing. This is purely because of their limitations in the computational abilities. Generally low cost tag security mechanisms are not expected to be bear relatively long attacks. If the attacks occur within short operating ranges it may be possible to detect them in a retail setting. Without relying on physical or electromagnetic properties, low cost tags should be able to resist attacks that last up to ten minutes. If this is possible then the tags are assumed to handle several numbers of requests per second.

Currently manufacture of thin tags is already in place and the cost of basic RFID tags is hoped to drop in the near future making it quite affordable to have tags attached to individual items as opposed to the pallets as it is currently. The reduction in cost is hoped to propel RFID tags into many areas of use. Some companies like Wal-Mart require that all its suppliers deploy RFIDs at the pallet level [8]. Another company Gillette is in process of embracing the RFID technology and has placed an order for tags for use in the supply chain and retail environments.

The possibility of RFID tags becoming ubiquitous also raises some potentially pervasive threats to consumer privacy. Tagged items will be subject to indiscriminate tracking if the RFID tags are not protected from unauthorised readers. This in turn leads to the tracking of the individuals as well who have the tagged items. But also companies can be in trouble as RFID tags can facilitate company espionage by revealing information about the operation of supply chains. This problem has been around for as long as the RFID technology has been and researchers are still working on finding a solution to the problem. There were some press reports that an apparel company Benetton withdrew plans for embedding RFID tags in its clothes due to the worries that the public heard RFID tags would pose.

Despite the current state of RFID technology or practices, certain aspects of the technology pose potential threats to individual privacy as we explain below.

Radio waves travel easily and silently through fabric and other materials and are not restricted by line of sight. Therefore clandestine collection of information can occur on items that have RFID tags embedded into them and subsequently the individuals in possession of the items. Furthermore since the tags can be read from a distance it is difficult for the individuals to know when and whether the tags are being scanned.

Using a combination of RFID and Global positioning system, it is possible to track an individual or a vehicle making use of the tagged items. Tracking of an individual becomes complete if the tags can be associated with the individual. For instance a tag embedded in apparel could serve as an identifier for the person wearing it. The information of tagged items may remain general but identifying things people wear associates them to particular events or groups.

RFID technology has the ability of having every item with its unique identity. As we mentioned earlier in the barcode system items have the same barcode irrespective of which part of the world they are. The use of this unique identity could lead to the creation of a global registration system in which every physical object is identified and linked to its purchaser at the point of transfer. By linking the identity with an individual then his purchasing habits can be determined thereby undermining his privacy.

For secondary use RFID technology can be used in creating profiles and tracking of movement of individuals or things. This would potentially have an impact on someone's employment or insurance if personal information such as medical prescription and health history is revealed.

Finally because RFID deployment makes use of storage facilities like databases for the tag data, it is able to facilitate the practises we have mentioned. This is true as most of the tag information could be later associated with personal identifying data as the database grows.

RFID technology has gotten the society concerned about their privacy and security [14, 15]. Amidst these worries research has been done and is still going on and it aims at finding the best ways to have RFID tags that are able to afford privacy and security but at a low cost. Some available protocols that have been designed to achieve both security and privacy are quite expensive to implement in RFID tags. Hoffstein *et al.* [16] propose a lightweight public-key cryptosystem called NTRU and Stern and Stern [17] also propose a lightweight digital signature scheme. While both of these lead to very efficient mechanisms compared to previously known public-key cryptosystems and digital signature schemes, they still require resources well beyond what is available on low-cost RFID tags.

Management of tag keys is paramount amidst the underlying mechanisms providing privacy and access control. It is there good to have the initialization, storage and transfer of keys economical. One thing is for sure that providing flexible access control and key management tools at a reasonable cost is really a big challenge at hand. Openness and flexibility of the design will be one of the ways of achieving the desired goals for low cost RFID tags.

6.3.1 Design assumptions

Low cost RFID tags are resource scarce devices. The design proposed by the MIT Auto-ID Center is limited to a few hundreds of bits of storage, approximately 5000-10,000 gates of which 250-1000 gates are to deal with security features and they have a maximum communication range of a few metres [9].

This kind of tag is expected to cost not more than $0.05 and given its resource constraint it will definitely be a passive tag. It is also expected that security protocols and computations must allow for read rates of hundreds per second. Power consumption may be another limiting factor given a particular implementation.

The limits for the low cost RFID tags are well below the requirements for a public key cryptographic system. Many symmetric encryption algorithms are also beyond available tag resources. For instance the Advanced Encryption Standard (AES) implementation has a range of 20,000-30,000 gates and this is way too high as opposed to the low cost design [10]. Also if we consider the hardware implementations of standard cryptographic hash functions like the SHA-1 or MD5 they are still very costly.

The assumption made is that the low cost tags have insecure memory whose contents can be extracted by say physical attacks. The good thing though is these attacks need physical contact with the tag and so they can be easily detected when done in public. RFID tags cannot be trusted to securely store long-term secrets when left in isolation.

The forward channel (reader-to-tag authentication) could be monitored remotely as opposed to the backward channel (tag-to-reader authentication) which could be monitored in a short range. But all this is dependant on the communication frequency between the tags and the readers. As a means of more authentication of the tags, readers may measure physical properties like signal power levels or response times. All these are incorporated in protocols in the bid to achieve the security properties which we shall introduce in the next sections.

We do assume a secure connection between the reader and the backend database. Readers are able to perform cryptographic calculations and interface with key management systems on behalf of the tags.

6.4 Anonymity and Privacy

In this day and age there are so many organisations that have come up with concerns about the emerging development in RFID technology questioning the security and privacy of the users of the technology. Due to the way RFID technology works that is silently and through radio waves, it is possible for it to be used in secretly tracking down some one or an item using the tags sewn into their clothing or the tags embedded into the items. Researchers have built up tiny beads invisible to the human eye and can be embedded in inks to be tagged to currency, DNA molecules or other products that law enforcement officers or retailers have a strong interest in tracking. This technology [12] is still under study but may get functional in future.

The steadily increasing use of RFID chips is most likely to leave no aspect of life free from the monitoring capabilities of retail, corporate and government organisations. Given all these worries amidst the advantages of the tags, some measures have been proposed in order to counter this relentless information gathering. Some of the measures include killing the tag after use probably by crushing, use of blocker tags which impair readers by simulating many ordinary RFID tags or specific tags (simulate only designated ID codes) simultaneously. Furthermore as a means of calming the consumers' worries, organisations have argued that most items with RFID chips cannot be tracked beyond an operating distance of about one and a half metres. This may have some truth in it but there is already a possibility of tracking goods in transit from the manufacturers to the stores.

Based on the growing worry of privacy and security of RFID, research has been done and is on going to see to it how consumer privacy can be protected and some protocols that have been proposed will be described in section 4 in of our study.

There are a number of definitions for anonymity. One of them is a condition in which an individual's true identity is unknown [13]. Another definition is that Anonymity is the state of not being identifiable within a set, called the "anonymity set". When referring to human beings, we say that a person is anonymous when the identity of that person is not known. Being anonymous is a result of not having one's identity, characteristics of significant features disclosed. The anonymity may rise from the fact that the person was not asked, as in an occasional encounter between strangers, or because the person is simply not willing to tell who they are.

6.5 General Overview

Protocols are designed to fulfil certain security properties like we mentioned earlier in the design assumption section. Many of the protocols achieve some of the security properties but not all. In our study we focus on the following security properties namely privacy, complexity, Clone resistance and Denial-of-Service (DoS) resistance. We shall briefly introduce some of these concepts.

Privacy has a wide range of definitions but in section 3 we come up with our security definition and we look at privacy in terms of not being able to trace a specific tag during authentication. This will keep the tag anonymous and thus the user thus protecting their privacy. Complexity is got to do with the way a given protocol is able to find a match with the keys in the backend database. Most of the protocols we look at in our study do an exhaustive search and others don't. Description of the searches is done in section 4.

Clone and DoS resistance is really about whether or not a tag can guarantee resistance against these attacks. Some of the effects of these attacks can be devastating so they need to be controlled. The cloning attack is when an adversary get a tag's responses then goes on to replay the respond to a legitimate reader. The effect of this can be felt when an adversary queries an item in a store that is of a low cost then replaces it for an item that cost higher thus the reader accepts the high cost item as a low cost item because the response is valid.

As for the DoS attack this is much to do with desynchronizing the tag with the backend database. This is mostly prone to protocols that use timestamps in the identification of tags. An adversary may send a timestamp that is grossly inaccurate and so that when the tag is queried it will respond with a message with a timestamp that is not within the range of those stored in the backend database. This will ultimately lead to desynchronization between the tag and the backend database and in most cases the tag looses its functionality.

More details about these security properties is given in section 4 and also explaining from our intuition and available information how some of these protocols achieve these protocols.

6.6 RFID Anonymity

RFID technology basically works on radio waves at a given frequency as mentioned earlier in section 6.2. Since RFID tags can be read at long distances, this raises a big question of how safe a user of the tags will be because anyone may be able to track them down with the help of the tags.

As a means of achieving anonymity cryptographic protocols have been proposed. These protocols help keep both the tags and readers safe because if any communication is to take place between the pair then the two have to authenticate each other. It is normally through a challenge-response mechanism. In some of these challenge-response mechanisms there is reader-to-tag authentication and tag-to-reader authentication and in others it is only reader-to-tag authentication. In any of the scenarios the tags are not supposed to reveal their identity to malicious or rogue readers implying that they have to authenticate the reader before they can disclose their identity. In this case the reader has to authenticate itself to the tag before it can get fully functional. So this is purely what goes on in authentication protocols.

Besides authentication protocols there are also collision avoidance protocols which are incorporated in all tags because of the fact that one single reader may send a signal at a specific time when there are a number of RFID tags in range. Because there are many tags in range there is no definite order in which the tags will respond as if they all responded at the same time then there would be a collision. So the collision avoidance protocols were developed to avert this problem. These protocols have a separate identifier which we shall refer to as collision avoidance key (ID) which could be independent of the data stored on the tag. The collision avoidance key (ID) is a globally unique and static collision ID which is prone to tracking. This is true especially if the adversary studies a collision detection behaviour pattern—they can learn the tag's identity.

Anonymity on the overall is hard to achieve in the collision avoidance protocol since the collision avoidance is embedded in the lower layer of the tag. Most of the tag manufacturers implement privacy-enhancing techniques in the higher levels making the tags in the collision avoidance protocol vulnerable to tracking.

Having made mention of the available protocols it is imperative that we point out here that the anonymity we are looking at is the one in the authentication protocols. Therefore we do not intend to make mention of the collision avoidance protocols again.

Anonymity in RFID technology is possible through the use of collision detection protocols and general authentication protocols. The collision detection protocols are also authentication protocols though they have the ability to detect collision during the time the reader queries the tags. When a collision occurs there could be a possible leakage of information from the tags and there is a passive adversary they are able to get the information and this would render the tags involved in the collision no longer anonymous. There are other protocols that do authentication but cannot detect collision detection.

In this section we have talked about RFID history and how the technology works mentioning some of its application areas and merits. We give the design assumptions of the tag and mention some of the issues that some potential users have and talked about some of them like for example cost and privacy.

7. The RFID Authentication Protocol

Authentication is a key in systems that are security dependant. This is true in the sense that if a user of a system has to be granted permission into the system then they have to prove that they are authentic and that they can use the system. This can be done by use of a password for online systems or smart cards for access control to building as well us some online systems.

Authentication can be done in two ways namely reader-to-tag (forward channel) and tag-to-reader (backward channel). During the reader-to-tag authentication the reader verifies itself to a tag as legitimate and in the tag-to-reader the tag verifies itself as legitimate. A tag is ready to offer its functionality to a reader after authentication of a reader-to-tag and also a system is protected from fake tags by authenticating a tag to a reader.

Authentication protocols can be categorized into two that is fixed access control and randomized access control. In the former category, a tag replies a reader with a fixed message and in the latter category, the tag replies to a reader with a pseudo-random message which varies each time of the responses.

The assurance of the identity of an entity at the other end of a communication channel is called authentication. There are a number of methods that are used to offer strong authentication and they can employ either secret-key or public key cryptography. The secret-key cryptography has the problem of exchanging the shared secret key. For public-key cryptography this problem is not there as the private key is kept secret in the signer's (sender) environment and more so the public key is published with a certificate.

7.1 Challenge-Response protocols in general

The verifier sends a challenge request to the claimant. The challenge can be a randomly chosen value which changes from one request to the other. The claimant confirms its identity by doing some work on the challenge using the secret key which is associated with the entity. At the receipt of the message from the claimant, the verifier goes on to check the message to decide whether the claimant knows the secret. If the communication between the verifier and the claimant is eavesdropped the information got should not provide any identification since the next challenge will be a different number. The challenge-response protocol may not be the best protocol to use especially if the goal is anonymity. This is because in the challenge-response mechanism one party has to identify itself to the other and verify that they are the ones they claim to be. However, they are necessary in an RFID authentication protocol where there is need to authenticate both the reader and the tag before any transaction can get underway. Since it involves disclosing the identity of the two parties then there may be no anonymity but in RFID authentication hashes and random values are used and so the parties involved that is the tag and the reader do not really have to disclose their identities but makes use of keys and random values and these are erased after every successful authentication session.

7.2 Unilateral and mutual authentication

Authentication where one party P is authenticated to another party Q is called unilateral authentication. This can be done by one-way or two-way challenge-response protocols. For instance, if we only have reader-to-tag authentication then that is unilateral. In the one-way protocol the timestamp mechanism is employed. P the signer sends the encrypted timestamp t_p to the claimant Q who decrypts it and verifies that the timestamp is acceptable as shown below.

$$P \rightarrow Q: E_k(t_p)$$

As for the two-way protocol, it makes use of random values. In this scenario the claimant Q must first send a random number r_Q to the signer P who encrypts it and sends it back. Validation is done by decrypting the response and comparing it with the random number sent as shown below

$$P \leftarrow Q: r_Q$$

$$P \rightarrow Q: E_k(r_Q)$$

This notion of authentication is borrowed and used in RFID authentication. However there is a variation as in RFID there is no encryption involved especially in the case of low cost tags because of the limited resources. The low cost RFID tags make use of hash functions and random number generators and of course with the help of the reader/backend database they can do the authentication. Some protocols afford only reader-to-tag authentication while other afford both reader-to-tag and tag-to-reader authentication.

7.3 System model

The RFID authentication system is made up of a reader R or there could be other readers depending on the goal of the system, a tag T_i ($1 \leq i \leq n$) and a backend database B and as pertains to privacy. Basically

Readers—Are devices with some cryptographic functionalities and used to interrogate tags. They are made up of an antenna which emits radio waves with signals to the tags in the vicinity. These readers are normally connected to the backend database and operate in real time as most RFID systems require that authentication is done in real time. We shall point out that occasionally we shall refer to the reader and backend database as the same entity. These two work hand in hand and in the coming sections we shall more than often say that the reader searches for the keys—in essence the reader is actually checking in the backend database as it is the one that stores the keys.

The backend database actually contains all the valid keys and other security information contained in the tags. The readers normally initiate the authentication process in the case of passive tags as the tags depend on the readers for power. The contents of the backend database vary from protocol to protocol some contain more data that others. For instance some have keys, random values, timestamps and hash look up tables as compared to others that have only keys. Overall depending on which protocol it is the backend database also contains other security information about the tag like its unique identifier and other manufacturing information which is not so relevant for our study. So we shall mainly focus on the keys, hash values, timestamps and random values in the protocols we shall discuss in the course of our study.

Tags—Are chips with antennas that receive the signal from readers and could have a hash function embedded in them. Tags are categorized into two passive and active tags. The passive tags are those have no power source and rely on the reader for their operation. They were devised as a way of coming up with low cost tags. Active tags on the other hand are those that have a power source which could be a battery or solar source and offer superior performance. In our study we focus on passive tags sine we are looking at ways of achieving privacy at low cost. Passive tags are constrained by resources like memory and therefore cannot afford public key cryptography (restricted computationally). Passive tags can have keys, hash functions and random number generators embedded on them and these can be used in the bid to achieve privacy but this still leaves the tags vulnerable to many more threats.

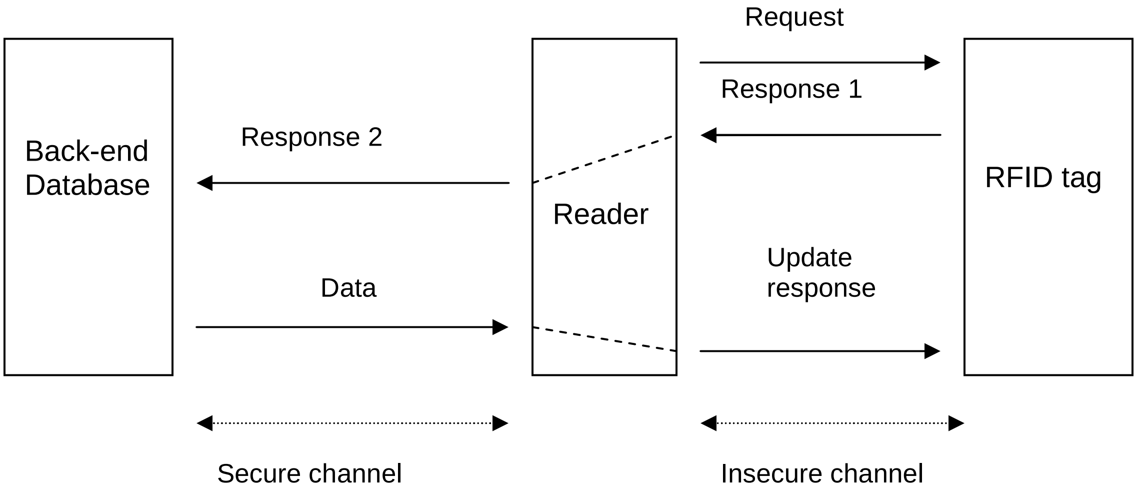

Figure 7 Illustration of a system model

RFID authentication protocols have been proposed to help in the authentication between the reader and the tag as a means of preventing unjustified tracking and monitoring, cloning and leakage of information. Some of these protocols are anonymous thus achieving privacy and others are not anonymous.

7.4 Adversary model

In this section we describe the various ways in which the adversary can execute some attacks. We also mentioned the communication channel that the adversary takes advantage of and also makes some assumptions on why we think the channel between the reader and the backend database is more secure compared to the channel between the tag and the reader. The model is made up of either making clone tags or malicious readers. The adversary also may be active or passive and for the former they are able to learn the conversation between a pair of R and T and can change the content arbitrarily therefore having the ability to terminate or initiate a session of choice. The latter on the other hand can be able to detect a conversation but has no ability to modify it.

Privacy and security concerns today lay their main emphasis on the backend database and the short range readers. The backend database is such a vital part of the RFID system which an adversary should not have access to. The communication channel between the reader and the backend database must therefore be kept secure to avoid any attacks from adversaries. Some RFID advocates hold the thought that an adversary without access to the backend database and with only short range readers is no threat. This is not true since an adversary is able to infer information from the communication between the reader and the tag and could come up with some meaningful information about the item which is tagged. This can be substantiated from the number of prevalent attacks on the communication channel between the reader and the tag. Since the communication channel between the reader and the backend is assumed secure we shall not focus on that but we shall turn our attention on the communication channel between the reader and the tag. By secure we mean that there is no threat from the adversary and if it is there is can be protected against. However, there could be a problem of privacy especially if a user for instance buys apparel from a store and goes back to the store wearing the apparel, the people of the store are able to identify the person and may slowly begin to gather information about what they shop most and how often they come to the store. This is a threat to privacy but the assumption we make is that the store will not undertake any of these actions against their customers. Furthermore we have decided to keep our focus on the channel between the tag and reader and address attacks the possible attacks between the reader and the tag. The assumption we make is that the adversary focuses their attacks on the channel between the reader and the tag which is normally insecure.

Considering an active adversary, there is a possibility that they are able to gather data from conversations between the reader and tag and then alter them to their advantage subsequently starting and ending a session at will. Active attacks are more likely to cause more harm to the system than the passive attacks. On the other hand the passive adversary who basically just eavesdrops is able to learn the output of the session by carefully studying the reader's decision to accept or reject a tag at the end of each session. There is no guarantee that the passive adversary can learn the content of the messages they see however the best they can realise is that there is communication going on between two parties in this case the reader and the tag.

With our earlier assumption that the channel between the reader and the backend database is secure which means that authentication is done before allowing any transaction, the adversary is not allowed to interact with the backend database expect only through a genuine reader.

7.4.1 Attacking model

An active adversary A can modify the conversations between any pair *T*, *R* adaptively and concurrently, and indeed initiate and terminate a session at its choice. As an extension of a passive adversary, A is also allowed to learn the output of the session. Since the channel between a reader R and the backend database B is assumed authenticated, we need not allow A to interact with the backend database B directly, but only through legitimate readers.

When designing secure RFID authentication protocols one should also take into account attacks that are excluded from the security model used and normally it is the system. There is a possibility that these attacks are prevented by using "out-of-system" protection mechanisms. But it is advised that you deal with such attacks within the model. We list two such attacks:

(a) Online man-in-the-middle relay attacks

These are attacks in which an unauthorized reader *R′* and the tag *T′* interpose between an authentic tag T and the reader R so that, the authentication flow in (*T, R, B*) is diverted to a flow (*T, R′, T′, R, TB*) that authenticates the imposter *T′* using the authentication data of *T*.

(b) Offline man-in-the-middle active attacks

These are attacks in which an unauthorized reader *R′* and tag *T′* step in between an authentic tag T and the reader R so that, when *R′* challenges T appropriately in (T, *R′*), the data obtained will leak private information of T when input to (*T′R, B*). In other words this can be known as a cloning attack.

In this section we started by giving a description of the RFID authentication protocol mentioning the available methods of authentication like public key cryptography. We went on to mention the system model with which we use to explain most of the protocols in our study and we also describe the adversary model. Finally we highlight the various aspects in which an adversary can carry out attacks in an RFID system. We have talked about the active and passive adversary and also mentioned what each of them is capable of. We also observed the fact that much as the backend database may affect the privacy of users especially if the items remain with the tags active. Much as there is an issue there we don't focus our study there but focus on the channel between the reader and the tag.

8. Security and Privacy issues

RFID technology is steadily getting deployed in pervasive computing and so that means that in not so long a period we may have the barcodes replaced with RFID tags. The merits of RFID tags as compare to the presently used barcodes are much more and so some big companies in the supply chain are steadily adopting them. In this section we give our security definition as well as mentioning some of the goals that anonymity authentication protocol should achieve.

8.1 Security Definition

There could be a number of security definitions but we shall focus on a situation where an adversary cannot trace tags. Being unable to trace a tag means that it is computationally hard to conclude from the interactions with the tag, information about the identity of the tag or to be able to link various manifestations of the same tag. We shall describe a circumstance where a number of tags can be queried.

Our main concern is that even if the so many tags are queried by an adversary A as shown in figure1 (left side), the adversary should not be in position to tell the differences between the tags $T_1$and T_2 but to recognize them as the same tag. Another scenario would be when the malicious reader (adversary) queries the same tag but at different times. Meaning that the malicious reader broadcasts a signal and tag T_1 responds then after a short interval it broadcasts a signal again and the same tag T_1 (figure 7 on the right) responds—the malicious reader should not be in position to identify tag T_1 as the same tag but rather receive information that is totally different from the information it received the first time round. If in the first scenario the adversary is able to tell the difference between T_1 and T_2 then the anonymity of the tags no longer exists and inevitably the privacy of the tags is at risk. In the second scenario if the malicious reader (adversary) is able to detect that it is the same old tag T_1 it has queried again then the tag's identity is no longer anonymous and therefore stands the risk of being tracked which is a privacy concern. On the overall tag output should be indistinguishable from truly random values and they should not be linked to the tag identifier.

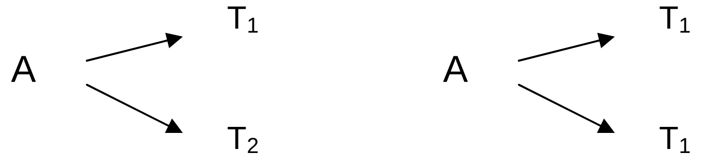

Figure 8 Adversary identification of tags

The other situation in which we have a security definition is the area of traceability. This comes in when a tag gets compromised or tampered with and the motive is to get the stored information in the tag which may contain the keys and the tag identifier. The adversary who tampers with the tag and acquires the information stored in the tag should not be able to trace the previous operations of the tag by studying the past events in which the tag was involved. Also a passive adversary should not be in position to connect the current output with the past output.

Satisfying the above security requirements is to use public key encryption but it requires more resources significantly and this increases the cost of the tag. So it is imperative that the above requirements are satisfied or met at low cost.

Basing on the above definitions it is clear that there are some technical issues that have to be dealt with to ensure that RFID tags secure against the possible threats. As we mentioned early one of the threats is the tag anonymity or privacy. If the tag's identifier can be kept anonymous then that solves the problem associated with the leaking of information about the tags' identifiers and subsequently that of the users of the tags.

Research work aimed at getting solutions to the possible threats privacy inclusive has been done and we shall describe some of the suggestions made in the next section.

8.1.1 Characteristics of the protocol

As earlier mentioned in section 7, a private authentication protocol should meet the following security requirements.

Privacy—Information such as tag's key, and other private information should not be leaked to any third party during the authentication process.

Untraceability—A tag's output authentication messages should not in any way be connected or form a pattern lest it may be tracked.

Cloning resistance—there should be no room for rogue tags to impersonate as valid tags. This is similar to a replay attack in which the tag response can be intercepted by a rogue reader then sent to a legitimate reader. This should be protected against.

Forward secrecy—A tag may be tampered with as a means of obtaining the keys stored in it. If a tag affords forward secrecy, those keys should not reveal any of the previous outputs of the compromised tag.

Compromising resistance—Privacy of tags is threatened if they share some keys with other tags. This is true in tree based approaches. As a means of reducing or controlling the effect of the compromising attack, the number of the affected tags should be minimised.

8.2 Security goals for anonymity schemes

The security goals that anonymity schemes have to achieve include the following

(a) Spoofing identity—In this arrangement an adversary can replace an authorized reader with a rogue reader thereby reading the tags in its vicinity without the owners' authorization. A good anonymity scheme should be able to offer mutual authentication so that both the tag and reader can be render legitimate before any transaction takes place.

(b) Modification of tag information—This can also be known as man-in-the-middle attack in which an adversary can alter the tag information in the bid to learn about other tags in the system. A good anonymity scheme should be able detect the inconsistence of the messages and so record and error in the authentication process thus protecting the tag from losing its functionality.

(c) Information Disclosure—An adversary may track a user and be able to determine where the user is located and where they have been by use of the tags carried by a user. A good anonymity scheme should allow for authentication before any transaction goes on.

(d) Forward Security—An adversary may decide to do physical damage to a tag with the intention of getting the information about the tag, What a good anonymity scheme should do is not to reveal the past operations of the tag. This will render the owner of the tag safe from traceability.

In this section we gave our security definition and it was based on traceability and identification. We also mentioned the goals that should be achieved by anonymity schemes.

9. Available proposed protocols that don't satisfy security issues

In this section we endeavour to give a detailed explanation of how some of the available proposed protocols operate and we highlight what they are capable of and what they are not capable of. We also explain the characteristics of hash functions and some of the terms related to tree based approaches and these are to do with complexity in key search. Most of the protocols we explain in our study make use of the hash based functions and tree-based approaches.

In this section as well, we come up with our contribution which is a description of the different security properties of the low cost RFID protocols we studied in the literature and how they are achieved. This we do in the security analysis section after each description of the protocol. By security properties we mean the security goals and other related abilities of the low cost protocols. As shown in the Table 1 we indicated which properties are met by the different protocols and these are namely privacy which in our case is synonymous to anonymity, forward security which is referred to as location privacy, clone resistance, DoS resistance, and complexity.

We shall future give a description of the properties we have listed above.

Clone resistance—by Clone resistant we mean that the protocol has the ability to protect a system against replay attacks. In certain circumstances a rogue reader may query a tag in order to learn its key so after getting the response from the tag it can be sent to a legitimate reader which will accept the response and subsequently the tag as valid and yet it has been compromised. This is what happens in a clone attack.

Privacy—earlier on in section we explicitly mentioned that we shall consider anonymity and privacy as synonymous. So this property basically indicates the ability of a protocol to keep a tag anonymous. We shall use it closely with untraceability since if there is privacy then we assume a tag cannot be easily traced. So in this property it is about keeping the information of the tag secure by preventing traceability.

Forward Security—this as earlier mentioned can be taken as location privacy. In a situation where a tag has been destroyed we expect that the adversary is able to get all the information contained in the tag memory. In some protocols the adversary is able to learn all the past transactions of the tag. So to be able to avoid this tracing of the past transaction a protocol has to offer forward security which is achieved by erasing all the previous information after each authentication session among other ways.

DoS Resistance—the DoS attack is especially aimed at either failing the entire RFID system or one of the components in the system. Often times the tag is targeted as it seems the soft target. In general DoS attacks cannot be totally done away with but the impact can be reduced. There are RFID protocols that are susceptible to a DoS attack in the form of a desynchronization attack which makes a tag loose its functionality. Therefore it is good for a protocol to have some resilience over such attacks.

Complexity—this is to do with the method which the protocol employs while searching for a key in the backend database. Some protocols with an unsorted list have to do an exhaustive or brute force search in which they have to check all the items in the backend database one by one until they get a match and this is referred to as linear complexity (search). Other protocols with sorted lists in the backend database can afford a binary search and so do not have to check the entire list of keys in the backend database but just half the list. This is faster and is referred to as logarithmic complexity (search).

After giving a description of the properties we shall go on to enumerated which of the properties are afforded by the protocols that we studied in the literature review. We realised that many of the protocols meet most of the properties but there is no single protocol that achieves all the properties satisfactorily.

Following we explain how each protocol achieves each of the mentioned properties or not.

Table 1 Properties of low cost authentication protocols

Properties(row)
Protocols(column)
	Clone resistant	**Privacy**	**Forward security**	**DoS resistant**	**Complexity**

Randomized Hash Locking scheme	Φ	Θ	Φ	Φ	Ξ
Strong Lightweight Protocol (SPA)	Θ	Θ	Θ	Θ	Ψ
Forward secure Protocol	Θ	Θ	Θ	Φ	Ξ
Light weight Protocol against Clone attacks	Θ	Φ	Θ	Θ	Ξ
Yet another protocol (YATRAP)	Θ	Φ	Φ	Φ	Ξ

Key

Φ ---- Not satisfied
Θ ---- Satisfied
Ξ ---- Linear
Ψ ---- Logarithmic

9.1 Hash functions and tree-based approaches

The definition of the hash function and the properties that continue after are based on the work of the Menezes, van Oorschot and Vanstones' book [25] , Steve Friedl [26] and Joux Antoine [27].

A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

9.1.1 Cryptographic properties

Apparently there is no formal definition that takes care of all the properties considered desirable for a cryptographic hash function. The properties that we list below are the ones that are taken as the fundamentals.

(i) Pre-image resistant—For all outputs y, it is computationally infeasible to find any input x such that h(x) = y given no corresponding input is known.

(ii) Second pre-image resistant—given an input m_1, it is computationally infeasible to find m_2 not equal to m_1 such that hash (m_1) = hash (m_2).

(iii) Collision resistant— it computationally infeasible to find any pair of inputs m_1 and m_2 such that hash (m_1) = hash (m_2).

A hash function may meet the above criteria but still have detrimental properties. For example, there is the length-extension attack to which most hash function are vulnerable. Without *m* and given *h(m)* and *len(m)*—length of m an adversary can calculate *h(m,m')* by choosing a suitable *m'*. This is property can be employed to crack naïve authentication schemes based on the hash function. However the remedy to this flaw could be to employ a keyed hash function otherwise known as a Hashed Message Authentication Code (HMAC).

There is a false impression that the "one-wayness" of a cryptographic hash function means irreversibility of processing a hash state. Such irreversibility in essence means the presence of local collisions that could make some attacks possible. The hash function must be a permutation processing its state bijectively (mathematical mapping) to be cryptographically secure. It should be irreversible regarding any data block just like any block cipher must be irreversible regarding the key. This makes iterated block ciphers and hash functions nearly identical when processing blocks of the same size as secret keys of those block ciphers. Most of the threats to the MDx and SHA families of hash functions utilize local collisions in the processing of the data block [28].

9.1.2 Complexity of Key search

Complexity theory deals with the relative computational difficulty of computable functions which is different from the computability theory which deals with whether a problem can be solved at all irrespective of the resources required. The complexity of an algorithm is the amount of a resource such as time that the algorithm requires. It is the measure of how "good" the algorithm is at solving the problem. The complexity of a problem is defined as the best algorithm that solves a problem. The time complexity of a problem is the number of steps that it takes to solve an instance of the problem as a function of the size of the input measured in bits using the most efficient algorithm. Let's consider an instance that is n bits long that can be solved in n^2 steps. In this case the problem has a time complexity of n^2. The exact number of steps will vary depending on the type of machine or language is being used. As a remedy to this problem the Big O notation is employed. This notation helps us generalize away from the details of a particular computer [29, 30]. We shall briefly look at the Big O notation as a means of trying to explain briefly the notion of the complexity of the key search. It is used in the analysis of the complexity of algorithms.

There are classes of functions that are normally used in the analysis of algorithms. All these are as n increases to infinity where n is the instance being considered. We shall focus on two types of complexities and these are Linear and Logarithmic complexity.

9.1.3 Linear complexity (Sequential search)

This involves a search algorithm that is suitable for searching a set of data for a particular value. It operates by checking every element of a list one at a time in sequence until a match is found. Linear search or Linear complexity runs in $O(n)$. If the data are distributed randomly on average N/2 comparisons will be needed. So since the list is randomly distributed (unsorted) there is the disadvantage of taking long to get an item. On the contrary, the simplicity of the linear search means that if just a few elements are searched it is less trouble than more complex methods.

9.1.4 Logarithmic complexity (Binary Search)

It involves a fast search algorithm for finding a given value in a sorted list. The Logarithmic complexity is denoted by $O(\log n)$ and is used to find an item in a sorted list with the binary search algorithm. Since the list is sorted it is expected that the item being searched will be got in short time. In other words it makes progressively better guesses and closes in on the sought value. For instance the binary search closes in on the sought value by comparing an element half way with what has been determined to be an element too low in the list and one too high in the list.

We shall illustrate an example of each of the above mentioned complexities. Painting a wall has a linear complexity since it takes double the time to paint double the area. Then looking up something in a database has a logarithmic time complexity since a double sized database only has to be opened one time more, that is, it splits in the middle and problem size is reduced to half.

9.2 Randomized Hash Lock scheme (Linear complexity)

In this protocol the tags are fitted with a hash function together with a pseudo random function (PRF). When the reader queries the tag, the tag will send a message, in which it generates a random value r, then hashes it with its key k and finally sends the combination to the reader. In essence the result is (r, h(k,r)) and r is chosen randomly. Figure 8 is an illustration of the protocol

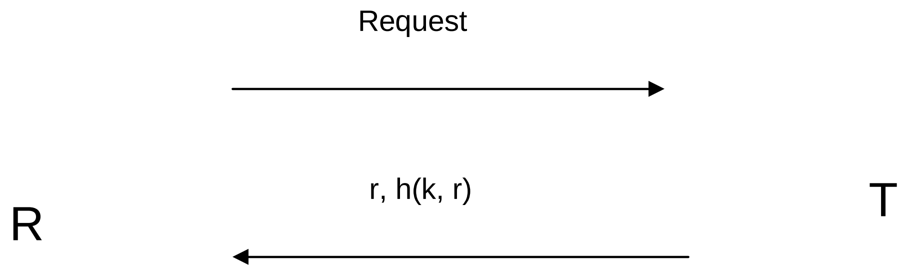

Figure 8 Randomized Hash Locking Protocol

For a reader to recognize the genuine tags, it must do a search through all the identifiers k_i that are stored in the backend database B. The reader achieves the following by performing an exhaustive search of all its known keys by hashing each of them together with r until it finds a match. After a match is found the reader can unlock the tag by sending the key value to the tag. On the other hand since the reader now knows the key value, it may leave the tag locked. This is illustrated in figure 9.

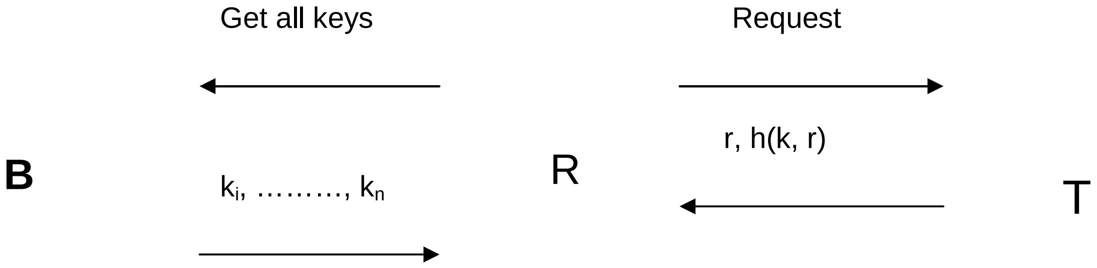

Figure 9 Identification of tags

The protocol employs a one way hash function as a means of making the previous output difficult to determine. The randomized hash locking scheme works well but has the disadvantage especially when it comes to the number of tags. If the number is large then it gets complicated as it uses a linear search (section 9.1.3) during the exhaustive search in the process of singulating (identifying) the tags. Given this setback we look at other protocols that overcome some of the setbacks and these are the secure RFID privacy protection scheme developed by Ohkubo et al. and the Strong lightweight RFID Private Authentication Protocol (SPA) developed by Li Lu et al [32].

9.2.1 Security Analysis

Cloning attack

In this scheme an adversary may send a request to a tag in the bid to learn or get the pair (r, h(k, r)). This can also be referred to as spoofing the reader to the tag. After spoofing the adversary can impersonate the tag by sending the spoofed values to a legitimate reader and in this case the reader identifies the tag and this is dangerous. The idea that the tag does not authenticate the reader makes this scheme vulnerable to the cloning attack.

Privacy

In the algorithm of this scheme we realise that during the time when the tag is responding to the request from the reader it sends a message (r, h(k, r)) with a random value included. The idea of the random value is to make sure that each time a tag is queried a different random value is sent and so this makes it hard for an adversary to infer any information about the tag. Secondly based on our security definition we can intuitively say that if a malicious reader queried the same tag again it would not be possible to tell if it is the same tag given that the response it sends entails a random value which is different each time the tag is queried. So we can conclude that privacy is achieved in this scheme.

Forward security

In this scheme the tag is susceptible to location privacy especially in circumstances where the tag gets destroyed as a means to get the stored information. This is true because as we mentioned earlier the scheme is prone to the clone attack and also that the random value is uniformly generated so if the adversary has been consistent in spoofing a tag to get the combination (r, h(k, r)) from the tag, they can closely study the information they have and easily relate it to the information they got from the destroyed tag thus revealing all the past transactions of the tag, subsequently not guarantying forward security.

DoS resistant

This scheme is capable of protecting a system against DoS attacks. Basing our argument on the idea what we described as one of the possible DoS attacks in section 8.2 we can say that since the tag does not refresh its key every after each query it is not possible that a desynchronization attack can occur. We notice that since the tag does not authenticate the reader any reader whether legitimate or not can query the tag and it sends its response (r, h(k,r)) which could be sent to the backend database and it will be considered valid. The adversary intentionally queries the tag so that it can send back as many responses as the number of times the adversary queries the tag. The adversary's notion here is to get the tag have a different secret key so that when a legitimate reader queries the tag, it cannot recognize it as valid. In this case we assume that the adversary thinks the secret key gets refreshed after each query. This is not true in this scheme thus protecting the tag from the desynchronization DoS attack.

Complexity

This scheme is not viable for owners of a large number of tags. This is because during the authentication process it does an exhaustive search of all the stored keys and this takes considerably a big amount of time to get done in a scenario where tags are many.

Normally if the search is done by checking against each of the items stored in a database then the assumption is that it is an unsorted list and that is referred to as linear complexity.

9.3 Strong lightweight RFID Private Authentication Protocol (SPA) (Logarithmic complexity*)*

This protocol addresses mainly two challenging issues on the reader and database side. One is the search algorithm (to reduce on cost of key search) in the key storage facility aiming at search efficiency as a way of supporting a large scale system and then two the security guarantee which is achieved by a dynamic update of keys.

Much has been done as a means of achieving efficient private authentication and so far tree-based approaches are arguably the most efficient protocols. This is derived from the notion that they use logarithmic complexity search schemes that are relatively efficient. SPA is one of the available tree-based protocols that we shall look at and it was proposed by Li Lu *et al.*[32]

Tree-base approaches may be efficient but are prone to security attacks in the area of key updating. Tree-based schemes mostly update their keys statically and not dynamically since the storage system is static implying that the keys are not updated after every successful authentication session. More so since the storage of the keys is in tree form then the tags more or less share common keys and this can be a potential danger if one of the tags is attacked and compromised by an adversary as it will leak information about the other tags. The figure 6 shows how tags T_7, T_8 share the same non-leaf node $k_{2,4}$ in the tree while T_5, T_6, T_7, T_8 share $k_{1,2}$. This sort of architecture is referred to as static and efficient since the complexity key search is logarithmic. For instance in the figure 10 below identification of any tag needs only log (8) = 3 steps.

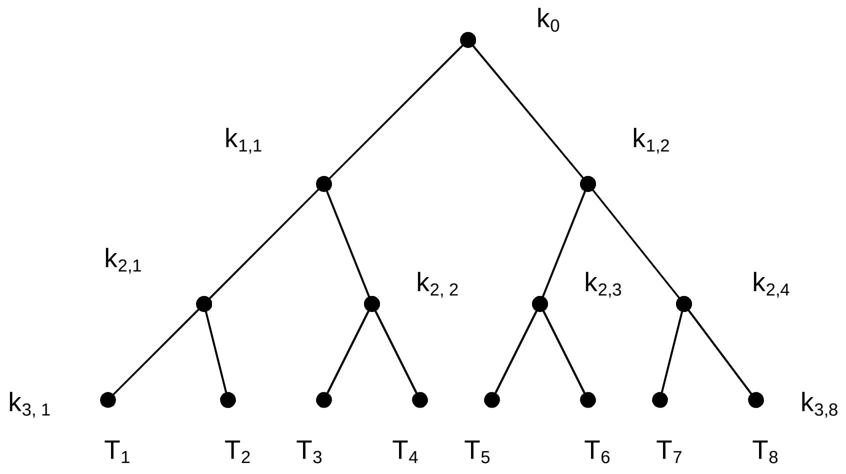

Figure 10 Tree representation with 8 tags

From figure 10 we realise that if an adversary is able to get a single key of any tag then they are capable of tracing a number of paths from the root to the leaf node. This is true in a static tree architecture where tags are not updated. This becomes an issue as the adversary is now able to differentiate the different tags from each other. If we are to go by our definition of privacy we concluded that it has been violated and therefore it is necessary that this problem gets fixed to solve the privacy problem. The solution to this problem is to be able to update the keys dynamically after every successful authentication session.

To ensure the consistency of the updating process, the keys of all the tags involved must be updated and redistributed. The trouble sets in when the key to be updated is closet to the root. This means that more than one half of the tags in the system are involved. One may think of getting all the affected tags and updating them in a specific period of time but this appears really difficult in practise especially if we are dealing with millions of tags. The difficulty associated with key updating led to the development of the SPA protocol.

9.3.1 SPA Overview

This protocol is made up of four sections namely the system initialization, tag identification, key updating and system maintenance. Like many other tree-based schemes, the first two sections are similar and basically do the authentication functions. We shall focus on the third section since it is the one that helps in achieving the security goals in the tree based approaches. The key updating is done after there has been successful authentication between the reader and the tag.

During the updating process, the shared keys between the reader and tag are updated while the validation used by the other tags is maintained and not broken. This is possible by use of two techniques that is the temporary keys and the state bits. The temporary key is for the storage of the old key for each non-leaf node in the entire tree and a number of state bits are used for each non-leaf node as a means of reading the key-updating status of nodes in sub-trees. As a result each of the non-leaf nodes will do their key updating after all its child nodes have updated their keys. This part of the SPA protocol guarantees the validation and consistency of the tags thus ensuring privacy. The authentication procedure of the SPA protocol is illustrated in figure 11 and the key updating in figure 12.

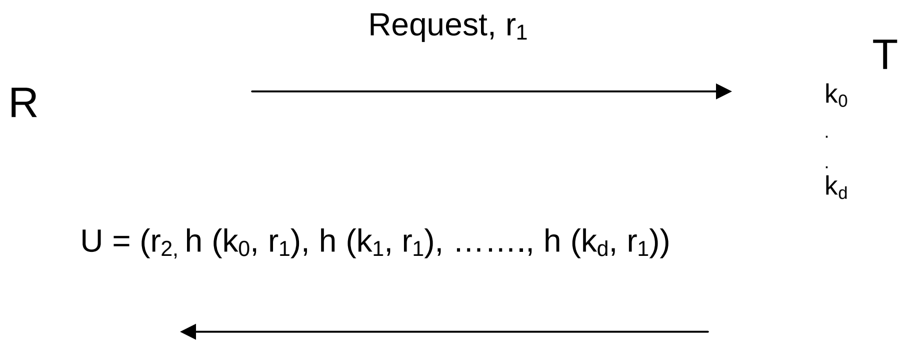
Request, r_1
R
T
k_0
.
.
k_d
$U = (r_2, h(k_0, r_1), h(k_1, r_1), \ldots\ldots., h(k_d, r_1))$

Figure 11 Authentication procedure in SPA

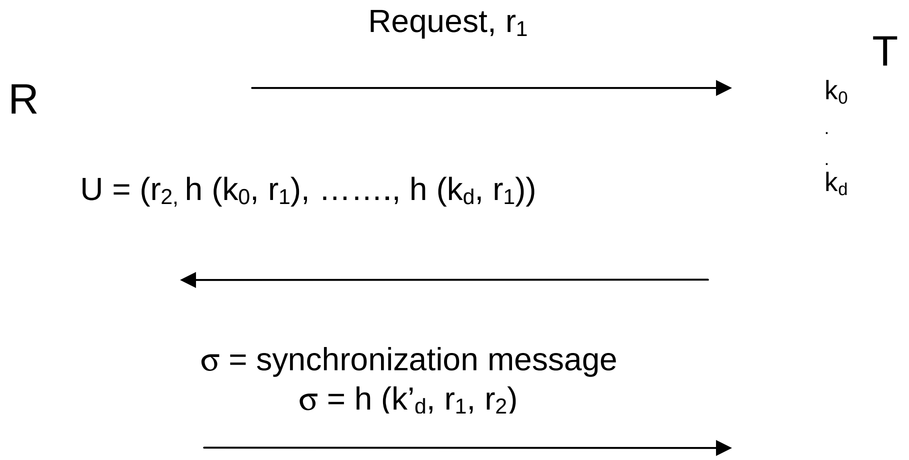
Request, r_1
R
T
k_0
.
.
k_d
$U = (r_2, h(k_0, r_1), \ldots\ldots., h(k_d, r_1))$
σ = synchronization message
$\sigma = h(k'_d, r_1, r_2)$

Figure 12 key updating in SPA

9.3.2 System Initialization

We shall consider a balanced binary tree to organize and store keys. The figure 13 shows a key tree with the branching factor $\delta = 2$ which we take as a binary tree. We also consider that there are N tags T_i $1 \leq i \leq N$, and reader R in the system.

The N tags are assigned to N leaf nodes by the reader R in a balanced binary tree S. A working key k_j and a temporary key tk_j are assigned to each non-leaf node in S. Each key is generated randomly and independently by the reader initially and so tk_j for all non-leaf nodes. When a reader sends a request to a tag T_i it distributes the ((log N) + 1) keys to T_i. These are the keys in the entire path from the root to the tag T_i. As we earlier mentioned that $tk_j = k_j$ initially at this point if $tk_j \neq k_j$ the tag T_i takes on k_j. In this case j represents a node at level ℓ in the binary key tree

S.

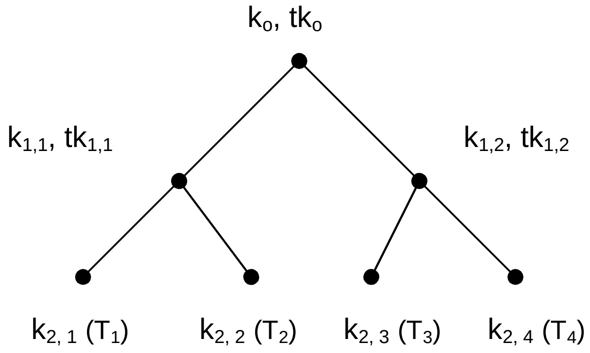

Figure 13 A key tree with N =4 tags

9.3.3 Tag Identification

The authentication between the reader and the tags is done in three rounds as shown in Figure 10. Firstly the reader R starts by sending a request and a random number r_1 to tag T_i $1 \leq i \leq N$. Then the tag T_i sends back a random number r_2 it generates and also computes the sequence $(h(k_o,r_1), \ldots,h(k_d,r_1))$ where h(k,r) denotes the output of a hash function on the key and random number r_1. So in essence the tag's response is $U = (r_2, h(k_o,r_1), \ldots,h(k_d,r_1))$. We shall represent the elements in U as $u, v_0, \ldots, v_d$. When the reader receives the response, the basic identification is executed as shown in figure 14 in the second step. The reader will start encrypting r_1 using k_0 from the root and compares the result with $h(k_o,r_1)$ from T_i. If a match is got R calls up a recursive algorithm in figure 10 to identify T_i. In the figure 13 the reader will begin from the root encrypting r_1 by using $k_{1,1}(tk_{1,1})$ and $k_{1,2}(tk_{1,2})$. If the result matches with $h(k_1^l, r_1)$ then the tag belongs to the left sub tree else it belongs to the right sub tree.

The reader computes all hash values $h(k_{l+1}, r_1)$ and $h(tk_{l+1}, r_1)$ by using the keys of node j's children then compares them with v_ℓ which is the authentication message U received from the tags T_i. If a match is found the reader continues the identification process until it gets to a leaf node. We realise that tag identification is tantamount to moving from the root to a leaf in the key tree as in the algorithm in figure 13.

```
Identification (U, mode n)
Fix the depth of the tree d by determining the log N;
  At this point the success is false;
Determine the depth of a node n and assign it to $\ell$;
 if ($v_\ell = h(k_n,r_1)$ or $v_\ell = h(tk_n,r_1)$)
   if ($\ell \neq d$)
        Record n in synchronization message;
        for i = 1 to $\delta$
         Find the children (n,i) of a given node and assign them to m;
          Identification (U,m);
  else if  $\ell = d$;
  The success at this point is true if all the conditions are satisfied;
 if (no the success)
     false so report a failure and output zero
else accept and output one.
```

Figure 14 Tree-based identification

9.3.4 Key Updating

When generating keys SPA still makes use of the hash function h. The reader computes a new key k_j' from the old key as k_j' = h (k_j). For consistence, the non-leaf node j uses temporary key tk_j to store j's old key. This avoids the interruption of the key updating of a tag and the authentication procedures of the other tags that belong to j's sub tree.

The reader R should update the keys of the identified tag T_i without interrupting the identification of other tags. There is also the idea that each non-leaf node should automatically update its key when all its children have updated their keys.

As a means of achieving the no interruption and automatic update of keys, SPA introduces a number of state bits to each non-leaf node. The mechanism works in such a way that each non-leaf node uses these bits to reflect the key updating status of its children. Whenever a child updates its key, the corresponding bit is set to 1. Finally each node updates its own key when all its state bits become 1. On completion of the updating process, the reader sends a message σ = h(k_d', r_1, r_2) and a synchronization message to T_i. The tag uses the message σ to authenticate R and the synchronization message is made up of information of the levels on which the nodes have updated their keys in the key tree. On receiving these messages T_i first verifies whether or not σ = h(k_d', r_1, r_2). If it is valid T_i updates its keys accordingly based on the synchronization message.

This algorithm guarantees that the key updating is consistent and feasible under arbitrary tag access patterns. The key updating algorithm is suitable for an arbitrary balanced tree with $\delta > 2$. In such a tree there are δ state bits maintained in each non-leaf node to indicate the key updating states of δ children. The figure 15 shows the key updating algorithm.

```
 Key updating (node n)
if (n is a non leaf node) then
 Store the old key as a temporary key
$tk_n \leftarrow k_n$ ;
 Thereafter generate a new key $k_n$ which is the hash of the old key
$k_n \leftarrow h (k_n)$ ;
Determine the parent of the child nodes and assign it to a value m
    $m \leftarrow$ FindParent (n) ;
if (n turns out to be the left child of m)
   set $s_m^l$ to one ;
      else if n is the right child of m
   set $s_m^r$ to one ;
if ($s_m^l = s_m^r = 1$)
   reset $s_m^l$ and $s_m^r$ to zero and record m in the synchronization message ;
if m is not the root node
 $n \leftarrow m$ ;
key updating (n);
```

Figure 15 Tree-based key updating

9.3.5 Security Analysis

Clone attack

Normally in cloning attacks the adversary (illegitimate reader) intercepts messages from a tag then tries to send them to a legitimate reader. In the SPA scheme the random value r_1 sent alongside the reader's query is used to give resistance against this attack. If the adversary used an illegitimate reader to query the tag in this scheme the tag will respond with the combination of U = (r_2, h (k_0, r_1), h (k_1, r_1), ……., h (k_d, r_1)) but if the adversary sends it to a legitimate reader, it will be regarded as an invalid response because the random values r_1 are not the same. So in this scheme the clone attack is not possible since the reader sends the request along with the random value r_1.

Privacy

In the SPA scheme, hash functions are embedded in tags and so we can say that with hash functions privacy is guaranteed. In case an adversary is eavesdropping on the communication between the tag and the reader, whatever information they will get will not be of any use to them or will not make any sense to them since it is just a random string of characters. Our reasoning of guarantying privacy is based on the notion that hash functions exhibit characteristics of randomness and one-wayness. We also assume an ideal state of the hash function that is not leaking any information that may be of importance to the adversary. In this way we can say that this scheme is able to protect the users' privacy.

Forward security

Generally speaking there is an attack that is not considered much because all tags irrespective of which protocol they are using are susceptible to—and this is called the physical attack and we can say it is one in which a tag is destroyed as a way of getting the information stored in it. In static tree based approaches it is possible for an adversary to get information stored in a tag after destroying it since the keys are not updated. For example in Figure 10 we have a binary tree in which we have four tags. In the static tree approach if tag T_4 got tampered with then it would mean that the adversary is able to learn the keys of the tags T_1, T_2, T_3 and by so doing the location privacy of the tags is not preserved.

In the SPA scheme, dynamic key updating is incorporated making it impossible for an adversary to get any information. Updating happens in such a way that after each successful authentication session all the non leaf nodes in the entire key tree get updated meaning that if one of the tag got compromised it would be hard to trace any past transactions of the tag or the tags that share some of the keys with the key tree. With this we can conclude that past operations of the tag cannot be accessible thereby guarantying forward secrecy.

DoS Resistant

This scheme is not vulnerable to a desynchronization attack. We support this by the following explanation. During key updating each non leaf node has got state bits which it sets to 1 or 0 depending on the success of the authentication. Besides the state bits there is what is called the temporary key and the current key. In the normal update process when the state bit is set to 1 then a new key k_j' is generated by hashing the current key $h(k_j)$. If the state bit is set to zero that means that the current key will be maintained and not refreshed. Before the entire tree non leaf nodes are refreshed each non leaf node maintains the old key as a temporary key as well as the current key. So if the update is not successful for the entire tree then the temporary keys are reinstated as the current keys. After the authentication session is done whether successful or not a synchronisation message is sent to the tag and it indicates the levels on the key tree that have been changed and so the tag synchronises its keys accordingly. With this ability of synchronisation in this scheme, it makes it safe from a DoS attack.

Complexity

This scheme is a tree based approach and in the tree based approach there is the concept of the branching factor δ in the key tree. We shall explain this using a binary tree. For a binary tree the branching factor $\delta = 2$ so when it comes to searching of keys we in essence have a sorted list and so a binary search algorithm is employed. In effective what is being done is the binary search. In the big O notation we have O(log N) which gives us the numbers of steps that the protocol will go through to get a match between keys or not. N in this case denotes the number of tags in the tree.

9.4 Forward secure RFID privacy protection scheme

Having talked about the randomized scheme and realising that it has some constraints in regard to the number of tags and the cost we take a look an RFID privacy protection scheme that is low cost and works well with a large number of tags. This scheme was developed by Ohkubo *et al.* [33] and is based on the review of previous work and suggests five issues that need to be considered in the design of an RFID privacy protection scheme. The five issues are about eliminating the need for unrelated rewrites of the tag information, eliminating use of high power computing units and minimizing the cost of the tag. It also has a priority of offering complete privacy and provision of forward security. Realisation of forward security requires the use of the hash chain technique to refresh the secret information in the tag. We assume that in the beginning the tag has information s_1 a random identifier stored in its memory and registered in the backend database B. Initially B contains a set of random values $\{s_i \mid 1 \leq i \leq n\}$. In the i^{th} transaction with the reader, the tag sends an answer $r_i = G(s_i)$to the reader and then refreshes the secret s_i to s_{i+1}. In essence $s_{i+1} = H(s_i)$. We must mention here that H and G are globally defined hash functions used for the transformation of the keys during each request or query. When the reader receives the tag’s response it sends it to the backend database. The backend database houses a list of pairs (ID,s_1) and here s_1 is the initial secret key and is different for each tag. During the search for a match, the backend database calculates $r'_I = G(H\ (s_i))$ for each s_i in the list and verifies if r_i is not equal to r'_i. On finding that $r'_I = r_i$ then there is a match and therefore it returns the identifier which is a pair of r'_i.

The idea they had is having the tag’s key refreshed every time a reader queries the tag and by so doing only the genuine users involved will be the ones to recognise the new keys. The scheme employs two hash functions, one to refresh the secret in the tag and the other to make responses of the tag untraceable by passive adversaries (eavesdroppers). The refreshment of the tag’s key happens autonomously by employing two hash functions G and H as indicated in the figure 16.

Following is the algorithm of the protocol is as shown

```
Reader                                  Tag
  r : = Request ();                          r: = receive (channel);
 send (channel, r);                          send: (channel, $r_i = G(s_i)$);
                                             Updates key $s_i$ to $s_{i+1}$ where
                                             $s_{i+1} = H(s_i)$
  x : = receive (channel);
 for i: = 1 to # db do
 if ($r'_I = G(H(s_i)) = x$)
 YES a match is found;
 Continue
else
 NO match found
 End (reject or error)
```

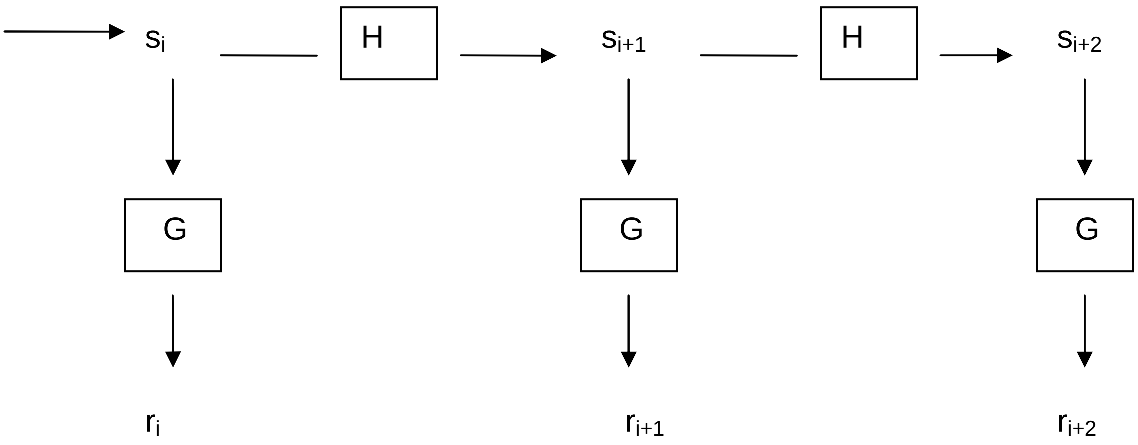

Figure 16 Updating the key in the secure RFID privacy protection scheme

9.4.1 Security analysis

Cloning attacks

The forward secure authentication scheme as we have explained has two hashes embedded in the tag and one is used to refresh the key and the other to update the key. What happens is when the tag is queried it responses with a value $r_i = G(s_i)$ then it updates the key s_i to s_{i+1} by hashing the previous key s_i. If an adversary tries to spoof the reader to the tag they will get the response $r_i = G(s_i)$ from the tag but if they send it to a legitimate reader it will be invalid since the key gets updated at each authentication session. Given the fact the key gets updated we can say this scheme is not prone to clone attacks.

Privacy

Privacy is about not leaking any secret information to a third party. In this scheme we two hash functions are employed and one of the hash functions is specifically to refresh the tag's key and the other one makes the response of the tag untraceable. In that way it makes it hard for an adversary to determine any relation between the messages r_i and s_i. For instance if the adversary gets the tag output r_i they cannot know s_i from r_i. Furthermore since the output of a hash function is normally a random value, the adversary would find difficulty in linking r_i to r_{i+1}. Privacy is guaranteed in that if an adversary gets the tag's response it will be difficulty for them to derive any useful information from the response of the tag given that the hash function generates a random string.

Forward security

As we are already aware this scheme employs two hash functions whose duty is to refresh the secret key of the tag and also to make the response of the tag untraceable. So even if an adversary tried tampering with the tag as a means of getting the key r_{i+1} or s_{i+1}, they cannot get r_i or s_i from r_{i+1} and s_{i+1} respectively. Subsequently they are unable to locate previous operations of the tag. This is clearly due to the effect of the second hash function whose role is mainly to make the response of the tag untraceable. We can conclude therefore that this scheme is practical for low cost RFID tags yet still observing location privacy in the event of tampering (forward security).

DoS resistant

In this scheme we have noticed that the secret key and the tag response are both refreshed after every authentication session. An adversary may decide to spoof the reader to the tag and get it response and since the tag does not authenticate the reader it will send its response each time it is queried. The adversary queries the tag and it responds with $r_i = G(s_i)$ thereafter s_i gets updated to $s_{i+1} = H(s_i)$. The adversary continues to query the tag with their illegitimate reader and so as he does so the key keeps getting updated. After a given number of queries the adversary will cease querying the tag and at this point the key is s_{i+a} where *a* denotes the number of spoof attempts. If a legitimate reader now queries the tag there will be a desynchronization between the tag and the backend database as the backend database will not be able to trace a match with $H(s_{i+a})$. Because of this it is clear that this scheme is not resistant to a DoS attack.

Complexity

This scheme also does a brute force search of all its stored keys to establish a match with any of the stored keys in the backend database and the response from the tag. So with this constraint we can say it works well for owners of a few tags. As we already mentioned that an exhaustive search involves some linearity this scheme also has linear complexity and so this brings in increases in the cost of the key search especially where we have a large number of tags.

9.5 A Lightweight RFID protocol to protect against traceability and Clone attacks

This scheme was proposed by Tassos Dimitriou [18] and generally depends on the following observation. If the identity of a tag changes in a secure manner after every read query then attempts like eavesdropping, spoofing, replay messages etcetera cannot compromise the security of the scheme. The scheme offers both tag-to-reader and reader-to-tag authentication and ensures forward privacy of transactions. It is also employs cryptographic functions to refresh tag keys and to make responses indistinguishable from random values. Dimitiriou proposed two versions of this protocol one after the other and this is the simple and enhanced protocol. We shall describe the simple protocol in this section and explain the enhanced protocol in the Denial of Service (DoS) section.

The two versions of this protocol are both dependant on the secret shared between the tag and the reader (back-end database). Generally speaking this secret may be common for all tags and in case one tag is compromised, then it implies that the entire system is compromised. This however can be solved by having many different secret keys for each tag. But still it has a bad side to it which is that a mechanism needs to be developed for the reader to determine which secret was used for which tag.

This scheme works in much the same way as the hash lock scheme only that it is free from the threat of the tag traceability since the tag identity changes with every application of the protocol. This is achieved by employing a secure one way hash function and random session identifiers in order to keep the tag responses untraceable.

The definition of a secure one-way hash function in this context is that it is one which is hard to invert and does not leak any information about the message on which it is applied. Initially the tag is loaded with a key k_o which is set to a random value.

The same data in the tag is contained in the database in addition to the hash value of the key which is the main identifier used to find an information related to a particular tag.

9.5.1 Initialization and Identification

The assumption made in this scheme is that during system initialization the tag is loaded with an initial key k_o which is set to a random value. The backend database contains the same data stored in the tag, combined with the hash value of its key, $h(k_o)$ that turns out to be the main identifier to look for any information pertaining to each tag.

During normal operation, the tag exposes no information apart from the hash of its key which is used to identify and address the tag. The identification is done as follows first the reader sends a request to the tag then in response the tag generates a new random value r and sends back ($h_k(k_i)$, r) and $h_k(r)$ to the reader.

When the database receives $h_k(k_i)$, it uses this value to search and recover the key k_i of the tag. If the database contains the k_i value then it can use the keyed hash $h_k(r)$ to complete the puzzle and verify the authenticity of the message. This step is so important since it prevents the cloning attacks by an adversary. With all these steps completed without error, the database accepts the tag as genuine and renews the secret key of the tag. The renewal of the secret key is done by hashing the current key to obtain a new key. The tag as well performs the same computation thus refreshing its key. It goes on to erase any relevant information like r and the previous key from its memory.

Below is an illustration of the protocol

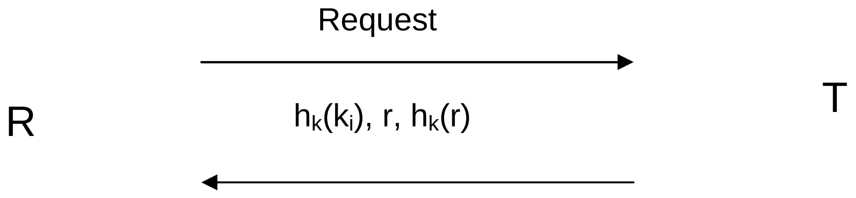

Figure 17 Dimitriou's lightweight RFID protocol

Following is the algorithm of the protocol

```
Reader                                              Tag
 r: = Request ();                            r : = receive (channel, r)
                                         send : (channel, $h_k(k_i)$, r, $h_k(r)$)
  x : = receive (channel);  for i : = 1 to # db do
  if $(h_k(k_i),r, h_k(r)) = x$
 then  YES there is a match
 else
 NO there is no match.
```

This scheme particularly achieves protection against tracing and forward security. This is true since each time the tag is queried the adversary is able to only see $h_k(k_i)$ and this will only be used once to identify the tag. Given the characteristics of a one-way function it makes it extremely difficult or totally impossible for an adversary to recover the original information and link k_i to k_{i+1} and also the use of the keyed hash function $h_k(k_i)$ protects the scheme against cloning attacks.

In a situation where the adversary gets access to the tag's memory it still becomes impossible to link the current key with the past actions since at each renewal of the secret key the previous is erased from the tag's memory. This guarantees forward security.

The success of this scheme is dependant on keeping the key shared between the tag and the backend database secret throughout the entire authentication process. In general if any protocol reveals its key then it is not considered an attack as it is obvious that without keeping the key secret then all the information about the tag can be determined.

9.5.2 Security Analysis

Cloning attack

In this scheme there could be a possibility of implementing the random value r using a timestamp and this is with the intention of preventing replay attacks**.** However, with a large number of readers there will be a problem of clock synchronization so it makes it hard to implement in readers as well as in the tags—since they have limited resources for the implementation of a clock.

In such a scenario, the adversary may send a request to the tag then in response the tag generates a new random value r and sends back $(h_k(k_i), r)$ and $h_k(r)$ to the adversary. In this case the adversary is an illegitimate reader. When the database receives $h_k(k_i)$, it uses this value to search and recover the key k_i of the tag. If the database contains the k_i value then it can use the keyed hash $h_k(r)$ to complete the puzzle and verify the authenticity of the message. The purpose of the keyed hash otherwise known as the message authentication code (MAC) is to authenticate both the message from the tag and its integrity without the use of any additional mechanisms. This step is so important since it prevents the cloning attacks by an adversary.

Privacy

In this scheme the tag increases it counter after every each successful mutual authentication session. But this could have an effect on the anonymity of tags. It may be assumed that this scheme offers privacy since it uses random values and hash functions but the adversary is able to query a specific tag several times so that the tag will be immediately recognized as it will respond with same messages. Therefore we can say that privacy can be achieved if the adversary is passive but it cannot be achieved if the adversary is active.

Forward security

For this property we build it on the basis of the explanation of the hash function that is it is a one way function. With the use of the hash function it becomes difficult to recover the original information and relate k_i to k_{i+1}. k_i is the shared secret key between the tag and the backend database and the security of this scheme is dependent on it. In this scheme after every successful authentication session all the previous information is erased from the tag's memory and what remains is the information about the current key. So in case the adversary tampers with the tag in order to get its current key it will be difficult to link the current key to past transactions of the tag.

DoS Resistant

This scheme has got two versions as indicated in the description. The first version was named a simple protocol and the second and improved version is called the enhanced protocol. The enhanced protocol was developed as a means of overcoming the desynchronization attack in the simple protocol. This attack requires the adversary to spoof the reader to a tag thus getting the response from the tag. As a matter of fact this message ($h_k(k_i)$, r, $h_k(r)$) is not of any use to the adversary but because of the algorithm of the protocol it will go on to refresh its key thereby having k_i become k_{i+a}. In this case a denotes the number of spoof attempts by the adversary. As a result since the tag ends up having a key that does not match the one in the backend database, the next query by a legitimate reader will lead to a desynchronization between the tag and backend database as the backend database attempts to find a match with the key $h(k_{i+a})$ in this case. This results in desynchronization and the tag loosing its functionality.

This problem is overcome in the enhanced version of this protocol by adding an extra message during authentication. Unlike in the simple protocol, we have the reader send a request along side a random value. So the reader constructs the message $h_{ki+1}(r_t, r_r)$ that will contain the new key and the two random values of the reader and the tag. This is then sent to the tag which generates its new key and then goes on to calculate the value $h_{ki+1}(r_t, r_r)$. If the value the tag calculates matches with the one received from the reader, then the response is accepted as genuine. At this stage the old key and r_t are erased from the memory of the tag. Otherwise an error occurs and the whole process aborts.

In this way the tag is protected from loosing its functionality and therefore we can say that this scheme gives resistance to the DoS attack.

Complexity

Many of the available protocols require doing a brute force search while finding a match between the response received from the tag and the keys stored in the backend database. This scheme is no exception as it also does the same. This search is costly in terms of time and besides that if the number of tags in the system increases it implies that more time will be required for the key search making it inconvenient in situations of a large number of tags. Since this scheme does an exhaustive search to find a key match the complexity is linear liked we mentioned in section 9.1.2.

9.6 Yet Another Trivial RFID Authentication Protocol (YA-TRAP)

It is a simple technique for inexpensive untraceable identification of RFID tags. The word untraceable here implies that it is computationally difficult to infer from interactions with a tag, information about the identity of the tag or link multiple manifestations of the same tag. This scheme is inexpensive since it requires one light weight cryptographic operation on the tag and also that it stores one key. It is specifically suited for the batch mode of tag identification. The notion of this scheme is making use of timestamps that increase or decrease steadily to provide anonymous tag authentication. The notion was a motivation from Herzberg *et al.* [34] in which they considered anonymous authentication of mobile users who move between domains. The domains may be a Kerberos secured internetwork or GSM cellular network. So it is all about a remote user identifying itself as an ephemeral (short lived) userid. This ephemeral userid is calculated as a collision resistant one-way hash of current time and a secret permanent userid. This also requires a backend database within the user's domain and it is contains a periodically updated hash table where each row corresponds to a travelling user. The row in the database is made up of permanent userid and an ephemeral userid. When a foreign agent sends a request to the database/ server, it looks up the ephemeral userid in the current table. The success of the lookup in the table is dependant on the fact that the timestamp used to calculate the ephemeral userid is recent. By so doing it protects the travelling user from being authenticated while avoiding any tracing by foreign agents or domains.

One of the merits of this approach is the database/server need not compute anything on demand as part of each request processing. All it needs to do is to pre-compute the current hash table and wait for any request. The cost of processing a request amounts to a table lookup which is significantly cheaper than a similar approach using random challenges where the backend database/server has to calculate an entire table based on the order of the requests to identify the travelling user. As the entire process continues the ephemeral table naturally expires and gets replaced with a new one. This is the main idea that was borrowed and employed in the YA-TRAP protocol for tag authentication.

The scheme by Herzberg *et al.* works well for mobile users but cannot be adapted to be used in the RFID system. This is true because most mobile devices are able to have clocks integrated within them and so are able to recognize incorrect time. For an RFID tag which has resource constraints, a clock cannot be incorporated and therefore it makes it hard for this scheme to be adapted in RFID systems as the tag will not be in position to distinguish between the correct and the incorrect time. But if the tag keeps the state of the previous timestamp considering it was legitimate then it can distinguish between future valid timestamps and past invalid timestamps. So with this observation the YATRAP protocol depends on the readers to offer supposedly valid timestamps to the tag at the start of the identification protocol.

The timeline in the YATRAP protocol is divided into small periods during which each tag should be authenticated once at most. The tags do not have clocks but are capable of storing a timestamp in their memory and also have a hash function embedded therein which is initialized by a secret key k and a timestamp t_0 and t_{max} while the backend database and the reader maintain a synchronized timestamp t_r. This protocol involves six steps of which the first three steps are crucial. The second step is made up of two rounds and two messages whose size is determined by t_r and $h_k(t_r)$ for the first and second messages respectively. The assumptions made in this scheme are that the channels between the reader and backend database are private and authentic and that the backend database only communicates with legitimate readers.

The reader sends a broadcast to the tag with the current timestamp t_r. If $t_r \leq t_t$ or $t_r > t_{max}$, then the tag broadcasts a random value else it broadcasts $h_k(t_r)$ and sets $t_t = t_r$. In this case t_t is the last timestamp that was received by the tag and $h_k(t_r)$ is the keyed hash function of t_r. When the backend database (reader) gets the tag's response, it looks up the value of the key that the tag used to generate $h_k(t_r)$ in the look up table. The hash table basically contains the time periods t_s, the keyed hash values and the keys.

When a timestamp t_s is updated in the table, the backend database calculates the keyed hash values $h_k(t_s)$ for all keys ($k \in \mathcal{K}$) as a means of getting the values for the next row in the hash look up table. The backend database depends on this table to find out whether the tag that issued the hash function is authentic that is $h_{tr,j}$ for $j \in [1,n]$ without having to do an exhaustive search for each time a new tag is challenged during the time period t_s which is normally one or a few minutes.

There are two modes of tag identification that is the real time and batch mode. Most of the considerations in this protocol are based on the real time mode. It involves immediate communication between the reader and the backend database as a means of quickly authenticating the tags in question. For instance in a library setup where books have to be monitored closely so that they are not stolen this mode is a must. Then in applications such as the inventory control where in most cases the readers are mobile and the items are stationary it would be better and cheaper to use the batch mode. In this way the reader could scan several tags get their responses and sometime later perform their identification in bulk. Generically batch mode is not the better mode in RFID systems since they are more of real time systems. This though may have its own demerits as well. For a batch of tags, the reader interrogates a group of tags, collects responses along with the corresponding t_r values to the server which goes on to identify the tags. This translates into O (n * log n) operations to identify n tags in other schemes [31, 33] whereas in this scheme it only needs O(n) since the same t_r specific hash table is used for all lookups and each lookup takes constant time. Overall this protocol is aimed at environments in which tag information is processed in batches rather than for more fine grained applications like access control and tagging of individual consumer items [39].

Following is the algorithm structure of the YATRAP protocol

Reader	**Tag**
r: = Request (timestamp);	r: = receive (channel);
send (channel. t_r);	send(channel,$h_k(t_r)$, random value)
y := receive (channel);	

```
for i: = 1 to # db
  if ($t_r$ == $t_t$)
 then
  $H_r = h_k(t_t)$ =y
 else if ($t_r > t_{max}$) or ($t_r \le t_t$)
 then $H_r$ = random value

YES a match is found
 Continue
 else
 NO match found
  Error
```

9.6.1 Security Analysis

Cloning attack

In this scheme timestamps are used and the decision to allow a tag as valid or not depends on whether a tag's timestamp is higher that the previous one. Now in this scheme the decision of the tag's response depends on the two conditions $t_r \le t_t$ or $t_r > t_{max}$. If any of the two conditions is fulfilled, then the tag will respond with a random value. On the next query of the tag there shall be a new random value assuming any of the same conditions still holds making the previous random value invalid. In this way if the adversary attempted a clone attack they would not be successful. Besides if the current timestamp t_r is equal to the tag timestamp t_t that is the other condition as shown in the protocol algorithm then the tag sends a value indistinguishable from the normal reply that is the keyed hash function to the reader thus protecting against the cloning attack.

Privacy

As we have just described previously that in this scheme uses timestamps monotonically increasing and this can be used by an adversary to identify a particular tag. He may not make the tag invalid but will know that it is the same tag they queried at one point. To be able to identify a tag, an adversary selects a timestamp t_f which will occur in future. The adversary goes on to queries the tag with the timestamp t_f causing the current timestamp t_r of the tag to become t_f. The timestamp t_f is in this case a unique identifier for the tag. The adversary can now start testing any time that is before t_f whether or not the unidentified tag is the same tag they gave the timestamp t_f. This whole process involves probing the tag and testing the results. During the probing two slightly spaced times t_f (before) and t_f (after) are chosen by the adversary so that t_f (before) $\le t_f <$ t_f (after). The next step is for the adversary to query the tag with t_f (before) and t_f (after) so that they can obtain r_f (before) and r_f (after) the responses from the tag. The adversary now interacts with the reader at time t_f (before) and t_f (after) and replays the responses r_f (before) and r_f (after). If the reader accepts r_f (before) and r_f (after) then they can more or less confirm that it is the same tag. In the real world an adversary can identify an individual on the basis of a single identified (marked) tag assuming they have many more other tags. So in this scheme privacy is not totally guaranteed.

Forward security

In this scheme an adversary is able to discover past transactions of the tag if it gets destroyed. The reason we say that this scheme does not afford forward security comes in from the fact that the adversary is capable of send a grossly inaccurate timestamp which they use to try and establish a specific tag as mentioned earlier in the privacy property and since the adversary is able to store some of the tag's responses then at the end of the day if the tag is tampered with to get the current timestamp then with the previous tag's responses with the adversary they are able to learn the past transactions of the tag thus in this scheme forward security is not guaranteed.

DoS resistant

As already explained to interrogate a tag, a reader transmits the current time t_r. The tag then compares t_r with the timestamp t_t of the tag. If t_r is out of date with respect to t_t then the tag outputs a random value. The use of timestamps creates some vulnerability to DoS attacks in the sense that the adversary could send an inaccurate timestamp t_{max} where t_{max} is the timestamp for the future. When the tag is queried it sets its internal timestamp t_t to t_{max} and then goes on to respond with random values in response to all future queries. This will lead to rejection of the tag in all the future sessions since the timestamp on the tag t_r is out of date. This is a DoS attack which makes a tag loose its functionality. Therefore we can say this scheme is vulnerable to a DoS attack.

Complexity

This scheme also employs the brute force search or exhaustive search while trying to find a match between the keys stored in the backend database and the response received from the tag. Since the search is exhaustive then the complexity is linear like explain in section 9.1.2. Using the big O notation we can denoted it as O(n).

In this section we have described the various low cost RFID protocols mentioning how they do the do the authentication. We have also laboured to give the whole process in an algorithm at the end of each description of the protocol. Furthermore Then we also did a security analysis for each of the protocols. The security properties achieved by each protocol are indicated in a table at the beginning of this section. Most of the RFID protocols as we noted operate in the real time mode since most system are needed in real time for instance in Libraries or supermarkets.

10. Conclusion of Part Two

In our study we have been able to talk about RFID technology almost from its inception up to where it is now in terms of development. However, we must say our focus has been really on low cost RFID authentication protocols and how they can achieve privacy given the resource constraints involved. As mentioned in section 6.1 RFID came into play around the time of World War II and it was developed roundabout the knowledge of radar which was invented by a British physicist. It then started by being used in the identification of animals especially during the time when animals needed to be treated before being transported or when there was need for vaccination of animals.

In the RFID technology we saw that the entire RFID system is made up of tags, readers and a backend database. In our study we assumed that the communication channel between the backend database and the reader is secure and therefore we focused on the channel between the tag and the reader. The RFID technology has also evolved from the time when any manufacturer was making tags according to their specifications and of course based on what needed to be accomplished. It is anticipated that RFID technology is likely to replace the Universal Product Code (UPC) which is commonly referred to as bar codes. This is simply because of the merits of RFID technology has over the Universal Product Code (UPC) like the ability to recognise an item with out line of sight like it is for bar codes and also the idea that each item can have its unique serial or identification number. These are really good in the supply chain but they also come with their own problems. Some of these problems involve cost of the tags which is still high and assuming all items or objects in the supply chain are tagged then it can be costly and therefore not economical. As a result there is research going on to make the technology low cost so that it can be used in pervasive computing. For this reason some companies that are already in need of using RFID tags have started using RFID tags but at the pallet level. Leaving the cost alone there are issues with privacy now that RFID technology does not need line of sight.

In our study we have described some of the protocols that have been proposed for low cost tags. We decide to grade them if you like and show the properties that they possess and how they can help the users be secure especially when RFID tags get deployed everywhere on almost all items. We realised that the available protocols do not achieve all the necessary security goals. Some are able to achieve a good number of the goals but if an adversary is determined they will use the flaw and track individual and know their habits like what items they normally buy and from which stores they shop. They could even trace their location especially if they have tags on their clothes or any other object that they move with wherever they are.

Most of the proposed protocols endeavour to hide their information that can lead to any discovery of the product on which they are tagged. So we realised as well that this is not a simple task given the limited resources and so we still see the need of research in the area of achieving a protocol that is able to achieve all the necessary security goals amidst the limited resources. Most of the suggested protocols capture a range of the special characteristics of RFID tag environment in a relative effective way. With this in mind we still feel there is room for refinement and improvement but this will partly depend on how the real-world embrace the technology and the way RFID tag systems evolve. Furthermore we say the development will rely more on empirical results from the areas in which the RFID tags have been used.

As we have mentioned that empirical results are key in the improvement of these protocols it is possible that even better protocols that may or may not hinge on the already proposed protocols may be proposed or developed. For further research we look at the areas of how RFID tags can be incorporated with more reason yet keeping the cost low. Many of the protocols look at the areas of using already manufactured low cost tags but the issue of exploring the development of low cost tags with more resources is still an open area for research.

11. References

1. The History of RFID Technology, www.rfidjournal.com/article/articleview/1338/1/129/

2. New report on RFID patents, www.rfidjournal.com/article/articleview/1338/1/129/

3. The EPCglobal Network™: Overview of Design, Benefits, & Security *Published* September 24, 2004

4. Dairy Farmers, Putting RFID in production, RFID Journal

5. Safe and secure Air cargo containers with integrated RFID Visibility System, RFID Journal

6. Konidala Divyan and Kim Kwangjo RFID Tag-Reader Mutual Authentication Scheme utilizing tags' access Password

7. Sanjay Sarma, David L. Brock and Kevin Ashton, The Networked Physical World-Proposals for Engineering the next Generation of Computing, Commerce & Automatic Identification

8. Walmart details RFID requirements, RFID Journal 2003

9. S. E. Sarma, S. A. Weis, and D. W. Engels. RFID Systems and Security and

Privacy Implications. In *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–470. Springer, 2002.

10. Martin Feldhoefer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong Authentication for RFID Systems using the AES algorithm, Institute of Applied Information Processing and Communication Graz university of Technology, Austria

11. Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. RFID Systems and Security and Privacy Implications. In *CHES*, pages 454–470. LNCS, 2002

12. Ari Juels, Minimalist Cryptography for Low-cost RFID tags, RSA Laboratories

13. Microsoft Computer dictionary Fifth Edition

14. S. Sarma, S. Weis, and D. Engels. Radio-frequency Identifcation: Security risks and challenges. *CryptoBytes*, 6(1), 2003.

15. A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In *Proceedings of the 7th Financial Cryptography Conference*, 2003.

16. J. Hoffstein, J. Pipher, and J. Silverman. NTRU: A ring based public key cryptosystem. In ANTS III (LNCS no. 1423), pp. 267–288, 1998.

17. Stern and J. Stern. Cryptanalysis of the OTM signature scheme from FC'02. In *Proceedings of the 7th Financial Cryptography Conference*, 2003.

18.T. Dimitriou, "A secure and Efficient RFID Protocol that could make Big brother partially obsolete," in proceedings IEEE PerCom, 2006.

19.A. Juels, "RFID Security and Privacy: A research survey," to appear in IEEE Journal of selected areas in communication 2006.

20.G. Avoine, E. Dysli, and P. Oechslin, "Reducing Tome complexity in RFID systems," in proceedings SAC, 2005.

21.T. Dimitriou, "A Lightweight RFID Protocol to protect against Traceability and Cloning Attacks," in proceedings SecureComm, 2005.

22. A.Juels, "Minimalist Cryptography for low cost RFID tags," in proceedings SCN, 2004.

23. M. E. Hellman, "A Cryptanalytic Time-memory trade-off," IEEE transactions on information theory, 1980.

24.D. Molnar, A. Soppera and D. Wagner, "A scalable, Delegatable Pseudonym Protocol enabling Ownership transfer of RFID tags," Workshop in selected areas in Cryptography, August 2005.

25. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography chapter 1.9 CRC Press 1996

26. An Illustrated guide to cryptographic hash functions by Steve Friedl

27. Joux, Antoine "Multicollisions in iterated hash functions. Application to cascaded constructions." LNCS 3152, 2004.

28. Cryptanalysis of MD5 and SHA: time for new Standard by Bruce Schneier

29. Michael Sipser(1997). *Introduction to the Theory of Computation*. PWS Publishing. Pages 226–228 of section 7.1: Measuring complexity.

30. Paul E. Black, "big-O notation", in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. 11 March 2005.

31. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems

32. Li Lu, Jinsong Han, Lei Hu, Yunhao Liu , and Lionel M. Ni Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems

33. Miyako Ohkubo, Koutarou Suzuki and Singo Kinoshita, Cryptographic Approach to "Privacy-Friendly" tags

34. A. Herzberg, H. KRawczyk and G. Tsudik, On traveling Incognito, IEEE Workshop on Mobile Systems and Applications, December 1994

35. Microsoft Computer Dictionary Fifth Edition

36. Kevin J. Houle, George M. Weaver, Trends in Denial of Service attack Technology 2001

37. Christy Chamon, Tri van Le and Mike Burmester, Secure Anonymous RFID Authentication Protocols

38. Vern Paxson, An analysis of using reflectors for distributed DoS attacks

39. A. Juels, S. Weis, Defining Strong Privacy for RFID, April 2006.

40. S.N Dorogovtsev, J.F.F Mendes "Evolution of networks", Advances in physics 6th March 2001.

41. Pearl, J. (1982) Reverend Bayes on inference engines: A distributed hierarchical approach. *Proceedings American Association of Artificial Intelligence National Conference on AI,* Pittsburgh, PA, 133—136

42. Kim, J.H and Pearl, J., (1983) A computational model for combined causal and diagnostic reasoning in inference systems, Proceedings IJCAI-83, Karlsruhe, Germany, 190-193

43. Radicchi, F.,Castellano, Cecconi, F., Loreto, V, and Parisi, D. "Defining and identifying communities in networks ", Proceedings of National Academy of Science in USA,101:2658-2663,2004

44. Newman, M.E.J & Girvan, M (2003), Cond-mat/0308217

45. Newman, M.E.J & Girvan, M (2004), phys Rev E 69:026113

46. Guimera` R, Sales-Pardo M, Amaral LAN (2004) *Phys Rev E* 70:025101(R)

47. Pan Hui, Eiko Yoneki, Shu-Yan Chan, Jon Crowcroft "Distributed Community Detection in Delay Tolerant Networks" MobiArch '07, Kyoto Japan.

48. A.Clauset. Finding local community structure in networks Physical reviews 2005

49. Holland, J.H "Adaptation in Natural and Artificial Systems" University Of Michigan Press, Ann Arbor. Michigan, 1975

50.Craig, James. (1981). Domestic Animal behavior: Causes and Implications for Animal Care and Management. Prentice-Hall;, Inc. Englewood Cliffes, New Jersey

51. Houpt, Katherine. (1991) Domestic Animal Behavior for Veterinarians and Animal Scientists. Iowa State Press, Ames, Iowa,

52. Mursel Tasgin and Haluk Bingol "Community Detection in Complex Networks using Gentic Algorithm"

53.C.Song, M. Havlin, and H. Makse. A self-similarity of complex networks. Nature, 433(7024)