DATA LEAKAGE AND PREVENTION SYSTEM IN SECURE CLOUD STORAGE USING LOCATION DEFINED NETWORK

Arulselvan A[1], Radhakrishnan B[2], Santhosh S[3]

Department of Computer Science and Engineering.
EGS Pillay Engineering College, Nagapattinam.

---***---

Abstract - Data sharing and access are capabilities businesses and organizations require the most these days. Remote working and mobile access to resources and collaboration platforms made it easier to access data and resources from anywhere, anytime. Employees want to access documents and email from different devices, and from various locations at a time. Access from untrusted networks is always a threat to businesses. This might result in data loss and overexposure of critical data. To mitigate the deficiencies of logical security mechanisms ,and coinciding with the trend of cyber-physical systems, security mechanisms have been proposed that integrate with the physical environment. To ensure that business's data and resources are safe. In this project we propose an innovative **CodeFence** that uses a location data and geospatial intelligence. Geospatial data analysis enhances understanding, insight, decision-making, and prediction. Location intelligence (LI) is achieved via visualization and analysis of geospatial data. Then we improve the security of data access in Data Server for a company or any other specific locations using the location-based cryptosystem. **CodeFence** provides a means to secure sensitive information within an organization. It can be set to Off, On, Restricted View or Read Only. Once a geo-fenced boundary is defined, the opportunities what businesses can do is limited by only their creativity. The main benefit of setting up such a geo fence is in avoiding data leakage. Once defined the trusted network locations, no one can access data from a different network location/device. The experiment shows that our scheme is feasible in practical applications.

***Key Words*: geospatial, data access, CodeFence, Location, Security, Cloud Storage.**

1.INTRODUCTION

In today's fast-paced and interconnected world, data sharing and access have become critical capabilities for businesses and organizations to thrive. However, with remote working and mobile access to resources come the risks of accessing data from untrusted networks and the potential for data loss or overexposure. To address these challenges, security mechanisms have been proposed that integrate with the physical environment, coinciding with the trend of cyber-physical systems. In this context, we propose an innovative CodeFence that leverages location data and geospatial intelligence to enhance the security of data access in a company or any other specific location. By using geospatial data analysis and location intelligence, CodeFence provides a means to secure sensitive information within an organization, with the ability to be set to Off, On, Restricted View, or Read Only. This solution allows businesses to limit the opportunities for data leakage and restrict access to data only from trusted network locations. Our experiment shows that this scheme is feasible in practical applications, and we believe that it has the potential to be an effective solution for enhancing data security in today's business environment.

2. System Analysis

2.1. Existing System

Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) scheme. STAMP is designed for ad-hoc mobile users generating location proofs for each other in a distributed setting. However, it can easily accommodate trusted mobile users and wireless access points. STAMP ensures the integrity and non-transferability of the location proofs and protects users' privacy. A semi-trusted Certification Authority is used to distribute cryptographic keys as well as guard users against collusion by a light-weight entropy-based trust evaluation approach.

- **xAd:**

This tool eliminates every form of assumption in marketing, because it serves messages based on your potential consumer's location. xAd has a proprietary platform that automatically creates boundaries around places often visited by a consumer. For example, a Restaurant, shopping mall. It's with these insights that marketers can target ads to their customers when they're within those locations.

- **Koupon Media:**

This tool prompts a targeted offer to shoppers when they're within the store. Koupon Media has features that study the behavioral attribute of buyers within the geofenced locations, and uses it to present the buyer with offers they can't resist while they are shopping.

- **NinthDecimal:**

This helps marketers to target consumers near their own stores or competitor's locations, with tangible media ads through phone calls, appointment requests, and couponing.

- **Wal-Mart**

It is another brand making it real big with geofencing. Their app comes with a store mode that picks up signals when a buyer is within the store, and delivers coupons and e-receipts.

2.2. Proposed System

This project will provide an introduction to Geo Server own authentication and authorization subsystems. such as from basic/digest authentication and CAS support, check through the various identity providers, such as Geo fence boundaries, MAC (Media Access Control), IP (Internet Protocol), as well as providing examples of custom authentication plug-in for Geo Server, integrating it in a home-grown security architecture. This system creates the victim file for wipe out the data, when the data is attempted to open outside of the geo fence.

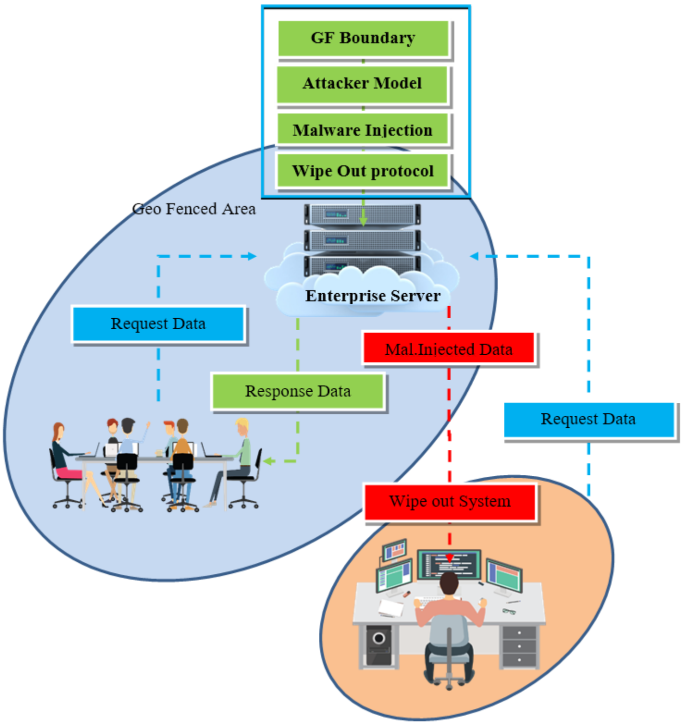
2.2.1. Virtual Fence

The project propose a Geo-fencing (geofencing) is a feature in a software program that uses the global positioning system (GPS) to define geographical boundaries. To check whether a person is within a geofence range we can make use of different algorithms such as Ray-casting, Winding Number, TWC (Triangle Weight Characterization) and Circular Geofencing using Haversine Formula. Geofencing is security, when anyone enters or leaves a particular area, an alert passes to server. This system creates the victim file for wipe out the data, when the data is attempted to open outside of the geo fence.

2.2.2. Geospatial Intelligence Technology

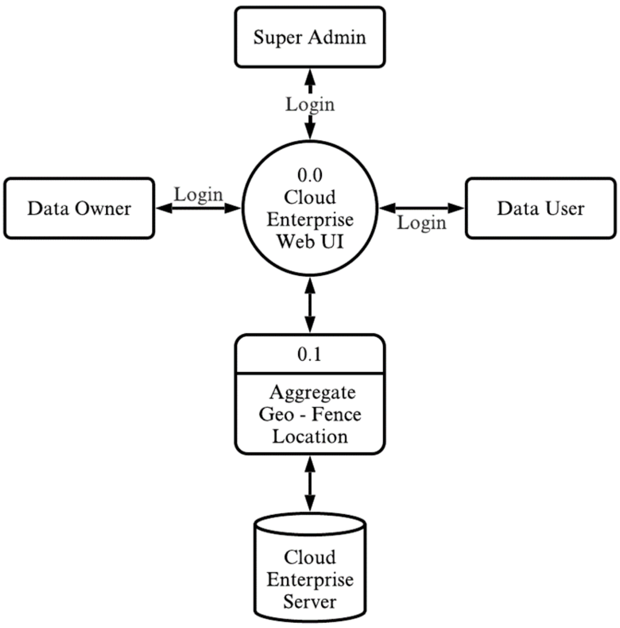
Geo-fencing (geofencing) is a feature in a software program that uses the global positioning system (GPS) or radio frequency identification (RFID) to define geographical boundaries'-fencing allow an administrator to set up triggers so when a device enters (or exits) the boundaries defined by the administrator, an alert is issued. Geofence virtual barriers can be active or passive. Active geofences require an end user to opt-in to location services and a mobile app to be open. Passive geofences are always on; they rely on Wi-Fi and cellular data instead of GPS or RFID and work in the background. Geofences can be set up on mobile, tablet, and even desktop devices anywhere in the world.

3. System Architecture

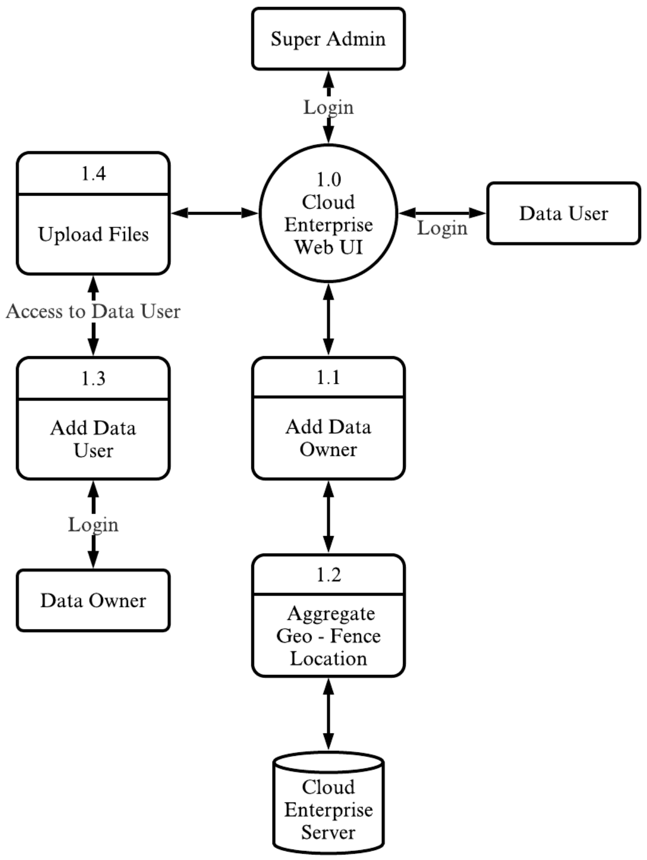


3.1. DFD

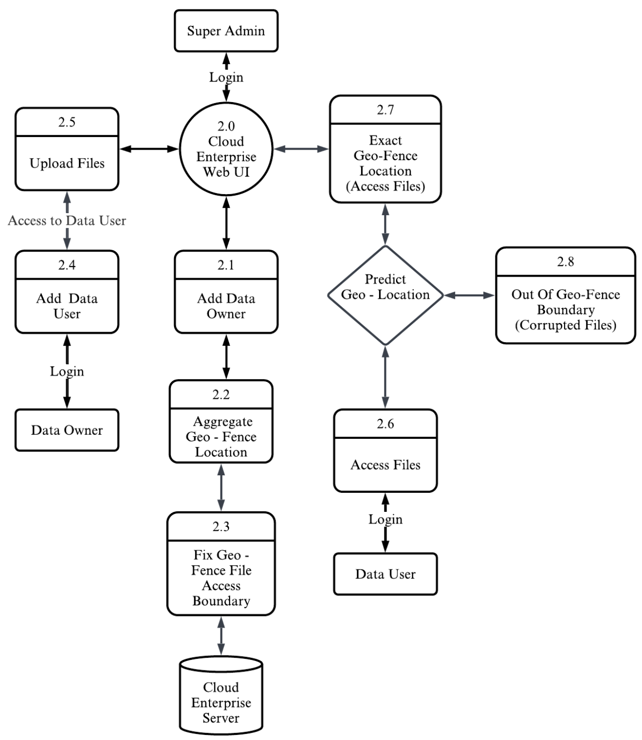
Level-0



Level-1



Level-2



4. Algorithms

4.1. Point in Geofence Framework and Algorithm:

Algorithm:

Input: **r** is the radius of the geofence. **g** = [g_o, g_i]
g_i is the position of the interest, go is the position of user.
Output: true if **r** does not violate **g**, otherwise false
if pointInGeofence (g_i, r) then
returntrue
end if
for all $g_{o(i)}$ in g_o do
    if pointInGeofence(r, $g_{o(i)}$**)** then
        returnfalse
    end if
end for
return true

The above algorithm consists of the input parameters r and g. r = (x, y) is the current position to check for geofence violation. The geofence is specified by g = [g_i, g_o] where g_i is the keep-in geofence boundary polygon and $G_o = \{g_{o1}, ..., g_{0n}\}$ is the set of keep-out boundaries. go_j is the j^{th} of n keep-out geofence boundary polygons. The PointInGeofence () function can be implemented by any of the algorithm like Ray Casting, Winding Number, TWC and Circular Geofencing using Haversine formula

4.2. Ray Casting:

The Ray Casting algorithm determines whether or not the position of interest, Gi, is inside a given polygon p, by projecting an infinite ray from Gi. If the infinite ray intersects an odd number of polygon edges, then r is contained in p, otherwise, r is outside of p. As the Ray Casting algorithm iterates over all edges of p and does not have an initialization step, if the geofence boundaries change from one time step to the next, code execution and results of the Ray Casting algorithm are not impacted.

Algorithm:

Input: p is a simple polygon
G_i is the position of interest buf
is a buffer distance.
Output: true if p contains **G_i**, otherwise false
count = 0
s is an infinite ray in the +y direction, originating at**G_i**
for all edges e in p do
    if **G_{i_x}** is within buf of ex then
        $e_{x,buf}$ = ex -2 *buf
    else
        e_{buf} =e
    end if
if **G_i** is within buf of e or e_{buf}then

4.3. Winding Number:

The winding number accurately determines if a point is inside a non-simple closed polygon. It does this by computing how many times the polygon winds around the point. The point is outside only when this "winding number" wn = 0; otherwise, the point is inside.

Algorithm:

G_i = a point of Interest,
V [] = vertex points of a polygon V[n+1] with V[n] = V[0]
n = number of vertices
Output:
winding_number
(when the winding_number =0, P is outside and winding_number is non-zero if P is inside)
PIP_windingNumber (Point G_i, Point V [], int n)
{
Wn=0:
1: int // the winding number counter
2: **for** each edge E[j]: V[j] V[j+1] of the Polygon **Do**
3: if (E[j] crosses upward)
4: if (P is strictly left ofE[j])
5: ++winding_number;
6: end if
7: else if (E[j] crosses downward)
8: if (P is strictly right ofE[j])
9: --winding_number;
10: end if
11: end if
12: end for
13: return winding_number;}

4.3. Triangle Weight Characterization (TWC):

Triangle Weight Characterization, consists of an initialization step and a run-time step as shown in Algorithm of TWC [4]. The initialization step must be executed for all keep-in and keep out geofences when the system first activates. If there are any changes to any of the geofence boundaries after the original initialization, each keep-in or keep-out geofence that is changed must be initialized again [4].
The initialization step subdivides each of the original geofences from simple polygons to y- monotone polygons and then to triangles [4]. The run-time step checks whether the position of interest is within each triangle. If the position of interest is inside any of the triangles, then it is within that polygon. Otherwise, it is outside the polygon [4].
Algorithm:

Input:
p is a simple polygon
$\mathbf{G_i}$ is the position of interest
Output:
true if p contains $\mathbf{G_i}$, otherwise false
Initialization:
1: Divide p to m y-monotone polygons
2: for all y-monotone polygons M in p do
3: Divide polygon M to n triangles
4: end for
Run-Time:
5: for all N triangles in p do 6: if N contains $\mathbf{G_i}$ then
7: return true
8: end if
9: end for
10: return false

4.4. Circular Geofencing Using Haversine Formula :

In the below algorithm, geofence of radius 'r' is created around the point Go The distance between Go and Gi is calculated using Haversine formula. The Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes [6].
Algorithm:

Input:G_i the position of the interest. Gi = [lat_i, $longi$] G_o
is the current position of the user. G_o [lat_o, $long_o$] r is the radius of the geofence.
Output:
True if the G_i is within Geofence range of G_o
checkWithInGeofenceRange ()
{
1: Distance = 0
2: Distance = getDistanceFrom Location (Go, G_i);
3: If (d < r)
4: Return true
5: Else
6: Return false
7: End if
}
Get DistanceFromLoaction (Go, G_i)
{
8: radius =6371; //radiusofearth
9: dlat = deg2rad (lati, -lat_o);
10: dlong = deg2rad (longi, -$long_o$);
11: a = Math.sin (dlat / 2) * Math. sin (dlat/2)
+ Math. cos (deg2rad(lato))
* Math. cos (deg2rad(lati))
* Math. sin (dlong/2) *Math. sin (dlong/2);
12: c= 2 * Math.atan2 (math. sqrt (a), Math.sqrt (1-a));
13: D= R*c;14: Returnd;}

3. CONCLUSIONS

In this project, we introduced a novel location-aware framework for provide data security, which enables the participation of workers without compromising their location privacy. We identified geo fencing as a needed step to ensure that data privacy is protected prior to workers consenting to a task. We provided heuristics and optimizations for determining effective geo fencing regions that achieve high task assignment rate with low overhead. It also generates the victim files; it automatically checks the geo - fencing boundary values and wipeout the system and files when geo -fencing and MAC Address is mismatch.

3.1 Future Scope

In the future, we plan to take into account more complicated policies to capture other privacy requirements other than the location. And also we insist this method to popular E-Mail Service Provider too.

REFERENCES

1. V. Rampérez, J. Soriano, D. Lizcano, and J. A. Lara, ``FLAS: A combination of proactive and reactive auto-scaling architecture for distributed services," Future Gener. Comput. Syst., vol. 118, pp. 56-72, May 2021.
2. R. Mokadem and A. Hameurlain, ``A data replication strategy with tenantperformance and provider economic prot guarantees in cloud data centers,"J. Syst. Softw., vol. 159, Jan. 2020, Art. no. 110447.
3. Y. Mansouri, A. N. Toosi, and R. Buyya, ``Cost optimization for dynamic replication and migration of data in cloud data centers," IEEE Trans. Cloud Comput., vol. 7, no. 3, pp. 705718, Jul. 2019.
4. A. E. Abdel Raouf, N. L. Badr, and M. F. Tolba, ``Dynamic data reallocationand replication over a cloud environment," Concurrency Comput.,Pract. Exper., vol. 30, no. 13, Jan. 2018, Art. no. e4416.
5. N. Mansouri, M. K. Rafsanjani, and M. M. Javidi, ``DPRS: A dynamicpopularity aware replication strategy with parallel download scheme incloud environments," Simul. Model. Pract. Theory, vol. 77, pp. 177-196,Sep. 2017.
6. C. Liao, A. Squicciarini, and L. Dan, "Last-hdfs: Location-aware storage technique for hadoop distributed file system," in IEEE International Conference on Cloud Computing (CLOUD), 2016.
7. N. Paladi and A. Michalas, "“one of our hosts in another country”: Challenges of data geolocation in cloud storage," in International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace &Electronics Systems (VITAE), 2014, pp. 1–6.
8. Z. N. Peterson, M. Gondree, and R. Beverly, "A position paper on data sovereignty: The importance of geolocating data in the cloud." In HotCloud, 2011.
9. J. Li, A. Squicciarini, D. Lin, S. Liang, and C. Jia, "Secloc: Securing location-sensitive storage in the cloud," in ACM symposium on access control models and technologies (SACMAT), 2015.
10. A. Albeshri, C. Boyd, and J. G. Nieto, "Enhanced geoproof: improvedgeographic assurance for data in the cloud," International Journal ofInformation Security, vol. 13, no. 2, pp. 191–198, 2014.
11. G. J. Watson, R. Safavi-Naini, M. Alimomeni, M. E. Locasto, andS. Narayan, "Lost: location based storage," in Proceedings of the 2012ACM Workshop on Cloud computing security workshop. ACM, 2012,pp. 59–70.
12. Y. Mansouri and R. Buyya, ``To move or not to move: Cost optimization ina dual cloud-based storage architecture," J. Netw. Comput. Appl., vol. 75, pp. 223-235, Nov. 2016.