Authors: Pavlo Nazarchuk
Status: Completed
Updated by:

RFC - The Anti Camera Protocol(ACP)

Abstract
The aim of this document is to discuss the ACP (Anti Camera Protocol).
A GPSLocation tracker protocol for cars.
The general concept is that all the cars connected to the server that are at a close distance should know each other's locations and sensitive information such as speed.

Table of Contents

1. Introduction

The aim of this protocol is to further improve security when it comes to driving. Through UDP requests being handled by a main server that will store and take care of the data packets sent by each car. It will keep track of location, speed information and precise distance. Any car closer than 10 miles to each other will know each other's location, preventing possible distractions and sight dependency. It can also be used to replace AI cameras that at times can be disrupted.

2. Conventions used

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL"

Note that the force of these words is modified by the requirement level of the document in which they are used.

1. **MUST** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

2. **MUST NOT** This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

3. **SHOULD** This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

4. **SHOULD NOT** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

5. **MAY** This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option **MUST** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option **MUST** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provided.)

3. Commands and Responses

Command:

0:

Purpose- This command is responsible for informing the server it is a car that was just turned on. The server **MAY** reject the request based on packet loss and authentication.

Parameters-

Car status - This is a boolean responsible for identifying if the car is on or off, being the request a mistake or purposeful.

Latitude - This is the geographical latitude location that comes with the car that **MUST** be updated every 5 seconds

Longitude - This is the geographical longitude location that comes with the car that **MUST** be updated every 5 seconds

Speed - Current speedometer information contained in each car. **MUST** be updated every 5 seconds

1:

Purpose- This command will inform the server that it needs to update the speed of the car. The server **MAY** reject the request based on packet loss.

Parameters-

Speed- Current speedometer information contained in each car.

2:

Purpose- This command is handled by the client. Upon receival the car **MUST** resend the information it tried to send. This happens when there is packet loss.

3:

Purpose: This command will inform the server that the car is being turned off. The server **MUST** get rid of all information related to the car in order to not cause any confusion to the other cars.

4:

Purpose: This command will tell the server to update the location of the car. This command **MUST** be sent by the car every 5 seconds in order to maintain the other cars updates on where every car close to them is.

Parameters:

Lat: This is the geographical latitude location that comes with the car gps.

Long: This is the geographical longitude location that comes with the car gps.

5:

Purpose: Request for public encryption key.

6:

Purpose: Public key share code.

4. Car identification

The port with which the car has sent the first authentication request will be used as identification

5. Body format (Client information handling)

Socket UDP request (Client side):

Message(String): Error code ! Boolean ! LAT ! LONG ! Speed ! CheckSum

First parameter:

0 - New car
1 - Speed update
2 - Checksum error
3 - Close connection
4 - Location update
5 - Encryption key request

Second parameter: Car Status (1 being on and 0 off)
Third parameter: Latitude
Fourth parameter: Longitude
Fifth parameter: Speed
Sixth parameter: Authentication code
Seventh: Checksum information

Server request handling:

Response(String):

{
IP:PORT,
Relative Distance,
Location: Lat/Long,
Speed
}

5.2 Body format (Speed update example)

Message: Error code ! Speed ! Checksum

- 1!50!Checksum

5.3 Body format (Car disconnection)

Message: 3 ! Checksum

6. Error handling

Errors will be handled server side such as authentication, checksum and unreachable clients.
When a client disconnects the server **MUST** handle the user not being reachable and should delete all information about the client itself. In case of a checksum error (packet loss) the server **MUST** handle that loss by re-requesting the information. To avoid further errors the server **MAY** send “connectivity checks” to users randomly in case of forced disconnection.
In case of authentication errors the user information **MUST** be ignored.

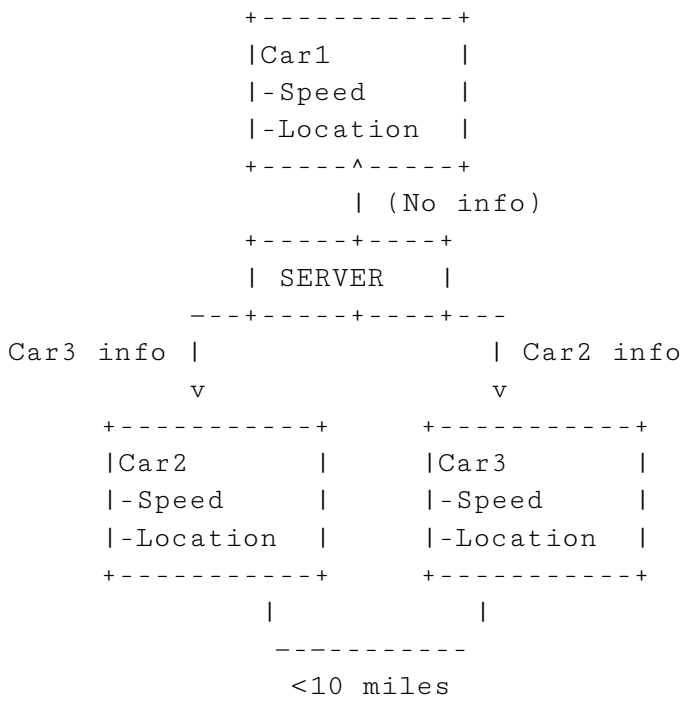
7. Security

DDOS will be handled by forcing the user to authenticate before having access to the storage of the server. RSA encryption is also implemented to avoid packet sniffing in case of the use of an insecure network.

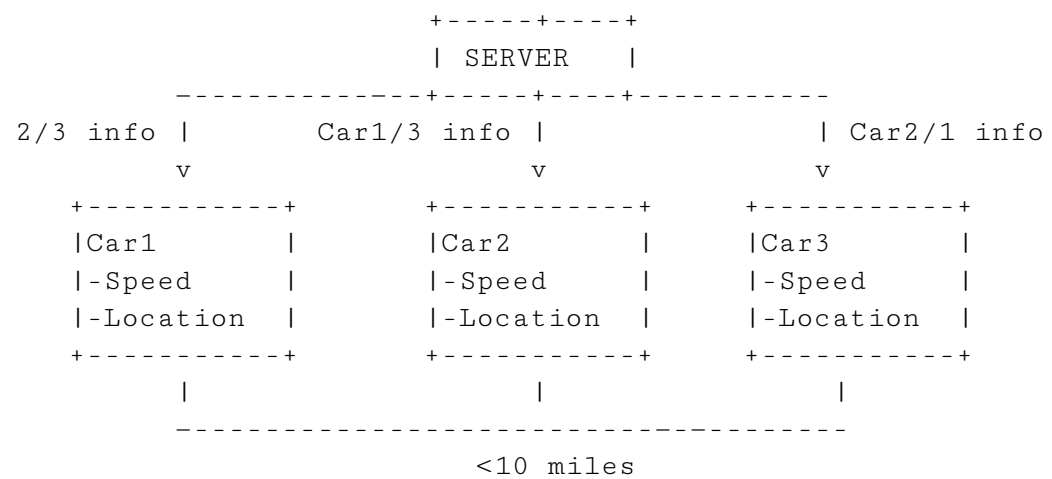
8. Data flow management

Requests will be handled on a FIFO basis. The users will send requests and until others handle it they will have a slight delay (ms).

9. Illustration



- Car 1 moves closer to car 2 and 3



10. References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>