
From: Andrea Amico, Founder and CEO, Privacy4Cars Inc.

Privacy4Cars Inc. ("Privacy4Cars") wants to thank the Colorado Department of Law for providing the opportunity to submit applications for universal opt out mechanisms to be considered for inclusion in its "public list of Universal Opt-Out Mechanisms that have been recognized to meet the standards of" the Colorado Privacy Act, pursuant to 4 CCR 904-3, Rule 5.07.

Privacy4Cars (https://privacy4cars.com) is the first and only privacy-tech company focused on the automotive industry. We have successfully convinced hundreds of companies, including vehicle manufacturers' captives, auto lenders, fleets and dealerships to start protecting vehicle users by deleting personal information (e.g., their home address, garage door codes, personal contacts, call logs, full text messages, geolocation history, and more) Consumers leave stored unencrypted in the vehicles they trade-in, return at the end of a lease or a rental, or are lost in an accident or are repossessed – including in Colorado. Even before we had our first paying business customer, we always gave our patented deletion tool for free to Consumers, so they can meaningfully exercise their Right To Delete. This summer we launched Vehicle Privacy Report (https://vehicleprivacyreport.com/), a first-of-its-kind privacy disclosure website that aims to educate the public about what data their vehicle manufacturer collects, shares, or sells, as well as which companies may have taken actions to protect Consumers. This tool is also free to use for Consumers, so they can meaningfully exercise their Right To Know. When Consumers told us how difficult it was for them to assert their privacy rights, we started filing thousands of Data Subject Access Requests on their behalf as their authorized agent – also for free.

Despite my company's and my personal efforts, privacy in automotive remains fundamentally broken. More than 4 out of 5 vehicles today are resold with the data of prior Consumers in it, and this data is leaked to dealership personnel, new owners, new renters, etc. on a daily basis because too many companies have not adopted reasonable safeguards and data sanitization standards (all stored data from all devices must be deleted) that are recommended by NIST[1], EPA[2], international industry standard bodies[3], and broadly used by retailers, refurbishers, and recyclers of virtually all other consumer electronics – but not cars. In our tests, less than 1 in 20 dealership associates could accurately explain to car-shopping Consumers basic facts about the privacy of vehicles they just test drove, including whether they captured personal information and if the manufacturers claimed rights to selling or sharing that information. A very large percentage of our Data Subject Access Requests go unanswered, or the Controllers' answers are late, incomplete, or not processed correctly. Meanwhile,

[1] https://csrc.nist.gov/pubs/sp/800/213/final

[2] https://www.epa.gov/smm-electronics/certified-electronics-recyclers

[3] https://e-stewards.org/learn-more/for-enterprises/overview/whats-the-e-stewards-standard/

many manufacturers state that just buying, or even just being a passenger in one of their vehicles constitutes "consent". Rental car companies similarly bury in their lengthy agreements that the act of renting constitutes "consent" to Consumers being surveilled. Manufacturers, rental car companies, and dealerships also frequently claim in their legal documents that it's the Consumers' fault – not the company responsibility – to protect the data their vehicles collect because Consumers "consented" to leaving it unprotected when they synched their phones and decided to not delete their own data despite the fact that vehicles give only very partial notices – do you want to download your contacts? – and as a consequence most Consumers are completely unaware that a lot more of their Personal Information, including very sensitive data, is collected, abandoned, exposed, and at risk.

For these reasons, when Global Privacy Control started to gain traction and states such as Colorado started to discuss having provisions in their privacy law drafts about Universal Opt-Out Mechanisms ("UOOMs"), I set out on a mission to develop a UOOM for vehicles. Using my cybersecurity experience of using modified smartphones to hack vehicles over Bluetooth, I designed a UOOM that leverages long-standing fundamental IT standards, which successfully passed our technical tests on the ability to turn on or off the signal, and to have it detected both by the vehicles and the apps that recent vehicles have. "OptOutCode" was born. I then realized that this UOOM works for most Bluetooth IoTs. It can also work for all IoTs that connect over Wi-Fi instead of Bluetooth. Soon after I discovered it also works for all apps (and app stores) that run on smartphones. As of 3 weeks ago it dawned on us that OptOutCode could be successfully used as a UOOM also on traditional computing devices (laptops, desktops, tablets, terminals) and all the programs that run on them.

The more we realized the power of our OptOutCode UOOM, the more we recognized the potential and the need to submit this UOOM to you. Global Privacy Control ("GPC") works well for interactions via browsers, but the average American has 22 IoTs in their home. They also spend four to five hours a day on apps, and several more on their computers or consuming digital media outside of browsers. GPC does not address any of it. We reached out and shared information with several advocates and academics to test the robustness and quality of our OptOutCode UOOM. Their feedback was overwhelmingly positive, and they encouraged us to write a technical spec (on which this document is based) and submit this application. Some of them – Electronic Frontier Foundation (EFF), Electronic Privacy Information Center (EPIC), and Surveillance Technology Oversight Project (S.T.O.P.) – have offered already their formal support. We are proud to submit for your consideration "OptOutCode".

## What is the name of the proposed UOOM?

Privacy4Cars is proud to submit "OptOutCode", a new Universal Opt-Out Mechanism. OptOutCode has four distinguishing advantages:

1. **It is truly Universal:** OptOutCode is compatible with smartphones, laptops, tablets, routers, the apps that run on them and the IoTs they connect to, including vehicles, smart appliances, tracking beacons, and more.
2. **It's fully decentralized:** OptOutCode does not require to build, maintain, query, or secure a central database of opting-out users, devices, or apps. The signal is detected locally: either by the programs and apps running on the device that has OptOutCode turned on (e.g., an app on a smartphone), or by other devices that connect to the device of the Consumer that has OptOutCode turned on (e.g. a SmartTV connected to a Wi-Fi router with OptOutCode on, a fitness watch connected to a smartphone with OptOutCode on). This architecture makes it privacy-preserving, more secure, without a single point of failure, cheaper, and faster.
3. **It's Consumer-friendly:** All Consumers need to do to turn on OptOutCode is simply rename their devices by adding "0$S" as the first three characters. For instance, rename their phone from "My Phone" to "0$S My Phone". Most Consumers can do it themselves by going through the settings of their devices in less than a minute, but we have also

written code to automate the task of turning OptOutCode on or off with a simple "switch" on an app.

4. **It's business-friendly:** Businesses can easily read and parse the "0$S" opt-out code from each device using backward-compatible and future proof protocols that require no special authorizations. We want to thank Professor Sebastian Zimmeck of Wesleyan University (credited with inventing GPC) for recently pointing out to us that Google has a somewhat similar protocol to allow users to opt-out of their Wi-Fi routers' signals[4] to be used by Google Maps, providing an independent industry validation that a portion of the name of a device can be used as an opt-out signal.

## What are the names of the individual(s) or organization(s) submitting this application?

Privacy4Cars is the sole submitter of this application.

That said, three high-profile non-profits and Consumer advocates decided to officially support our application after meeting and discussing/reviewing the technical aspects and reasoning behind our UOOM proposal:

- Electronic Frontier Foundation (EFF)
- Electronic Privacy Information Center (EPIC)
- Surveillance Technology Oversight Project (S.T.O.P.)

## Please provide a general overview of the UOOM, in plain English.

OptOutCode simply requires a device owner to modify the name of a device by adding the standardized prefix "□$S". For example, the owner of a smartphone would opt-out of the sale or sharing of their personal data collected by their phone, by the apps running on their phone, and by the IoTs connected to their phone by simply turning OptOutCode on their phone, e.g., by changing the name of their phone from "My Phone" to "□$S My Phone." In order for it to work, OptOutCode must be turned on in at least one of two devices that are paired wirelessly, or on the device that runs apps or other software. For all tense and purposes, Consumers should be able to opt out from most if not all "Targeted Advertising or the Sale of Personal Data" by renaming three devices: their smartphone, their personal computer, and their home router.

We settled on the prefix "□$S" for a number of reasons:

A. **It's memorable:** "0$S" is short for "do not" (zero) "sell" (dollar) "or share" (capital letter S). It's also a tongue-in-cheek commentary on how much companies should be able to financially benefit from selling and sharing the data of devices when OptOutCode is on (zero-dollar-S).
B. **It's short:** it requires only three (3) extra characters, or four (4) extra characters including a space or underscore to separate the OptOutCode from the original device name.
C. **It's uncommon:** searching online for "□$S" yields no results and the acronym is also not easily confused with initials or other meanings.
D. **It's easy to detect:** adding a prefix is preferrable to appending a string after the device name because it grants standard positioning (first three characters of name) hence is easier to parse with code even with devices with low computation power (e.g. some IoTs). A prefix is also

[4] https://www.androidpolice.com/2020/05/03/opt-out-wi-fi-from-google-location-services/. See also, https://support.google.com/maps/answer/1725632?hl=en.

more visible when a human reads a list of connected device names (for instance on the display of an IoT), making it easier to sort and/or visually confirm the request to opt-out.

We have provided several illustrative examples of how OptOutCode can work to express the Consumer's desire to opt out in a variety of scenarios in our technical documentation (which will be uploaded to https://optoutcode.com, a micro site currently under construction), but here are a few illustrative ways in which Consumers can opt out:

A. **A Bluetooth IoT connected with a smartphone:** if a Consumer were to pair their smartphone that has OptOutCode on with a Bluetooth Classic/BLE device (e.g. a vehicle, a fitness band, a Bluetooth tracker), that second device can, during the original pairing and every time it comes back in range of the user's smartphone, read the name of the smartphone (a standardized feature of wireless protocols), parse the first three letters, and if they are "0$S", it can interpret it as a signal that the user wants to opt out.
B. **An app running on a smartphone:** if a Consumer downloads an app (for instance, a game or a social media app) on a smartphone that has OptOutCode on, the app can read the name of the smartphone it is installed on without requesting special permissions, parse the first three letters, and if the name of the smartphone starts with "0$S", it can interpret it as a signal that the user wants to opt out.
C. **A Wi-Fi IoT connected to a Wi-Fi router:** if a Consumer changes the name of their router to start with "0$S", hence turning OptOutCode on, and connects a Wi-Fi device (say, a home speaker or a Smart TV) to that router, the device can read the name of the Wi-Fi, parse the first three letters, a and if they are "0$S", it can interpret it as a signal that the user wants to opt out.
D. **An application/software running on a laptop/PC/terminal or other traditional computing device:** if a Consumer is running software (for instance, their email program, or a browser) on a PC that has OptOutCode on, the application can read the name of the computer it is installed on without requesting special permissions, parse the first three letters, and if the name of the computer starts with "0$S", it can interpret it as a signal that the user wants to opt out. Specifically in the case of a browser, having OptOutCode on the computer may be a signal that the browser interprets to turn on GPC, hence propagating the user's desire to opt-out from the program (the browser) to the websites it visits (via GPC).
E. **A Bluetooth or Wi-Fi "sniffer"**: Many businesses deploy hardware (e.g., in retail stores) that listen to, locate, and log the Bluetooth and Wi-Fi signals our smartphones silently broadcast at all times. This monitoring happens even without the Consumer actively pairing their device. If a Consumer turns on OptOutCode by renaming their phone to start with the letters "0$S", the beacons/sniffers can read that name, parse the first three letters, and interpret it as a signal that the user wants to opt out.

It is important to notice two things: (1) the timing of these signals matter, as it offers both Consumers a way to simply opt out and companies a way to offer an opportunity to those Consumers to opt back in, and (2) how the opt-out signal should be interpreted varies slightly depending on the situation. IoTs, specifically, have special considerations due to a combination of one or more of the following factors: limited computing power, the fact that manufacturer and Controller may not be the same entity, they can store sensitive Consumer data with little or no security, and they can exchange hands across multiple users and owners.

### (1) Timing of the opt out signals, and ways to opt back in:

When a Consumer activates OptOutCode on a device, it is their "affirmative, freely given, and unambiguous choice to opt out of the Processing of Personal Data". Since Consumers can give and revoke consent at any time, every OptOutCode detection should be considered as an override to any previous consent given by the Consumer. For example, a Consumer may purchase a vehicle or download an app, and the moment they sign the sale contract or tap that "OK" button is considered by companies as the Consumer consenting to the processing of data they disclose in their Terms of Service and Privacy Policy. However, if at a later time that same Consumer turns on OptOutCode, or syncs a

phone with OptOutCode turned on, or launches the app on a device with OptOutCode on, that new UOOM signal should override any previously given consent.

By the same token, companies who detect an OptOutCode could decide to prompt the user (e.g., through a pop-up inside the app, a notification on their phone, a message on the display of the IoT) and offer them a way to opt back in, and, if the Consumer chooses to do so, this later signal should override the opt-out-signal of the OptOutCode.

In other words, it is still possible for Consumers to agree to their data being used (e.g., for marketing purposes or loyalty programs) but in order to do so companies must get the Consumer's affirmative, freely given, and unambiguous opt in after the OptOutCode signal is detected.

(2) How the opt-out signal should be interpreted (and why IoTs are special):

When a user expresses their opt-out by turning on OptOutCode by adding the prefix "0$S" to their device name, companies should interpret it differently depending on the type of data collection and device type. This is mainly because many IoTs have limited computational capabilities and/or the "Controller" may not be the manufacturer of the IoT but the company that owns it/has a financial interest in it and deploys it, many IoTs can store sensitive data but lack or have low security, and because IoTs more frequently are used by different data subjects and/or exchange hands.

A. For apps and programs that collect and transmit Consumer data and are running on devices with OptOutCode on, OptOutCode should be interpreted as a request from the Consumer to the data Controller (typically the publisher of the software, but in some cases it may be the company using the software) to opt out of the Processing of Personal Data for purposes of Targeted Advertising or the Sale of Personal Data pursuant to C.R.S. § 6-1-1306 (1)(a)(I)(A) or (1)(a)(I)(B). Examples of this category include smartphone apps and computer programs.
B. For IoTs that collect and transmit Consumer data either through an independent connection or leveraging the connection of a device that has been connected to the IoT with OptOutCode on, OptOutCode should be interpreted as a request from the Consumer to the data Controller (which could be the IoT manufacturer and/or a company that bought/leased/licensed/deployed the IoT) to opt out of the Processing of Personal Data for purposes of Targeted Advertising or the Sale of Personal Data pursuant to C.R.S. § 6-1-1306 (1)(a)(I)(A) or (1)(a)(I)(B). Examples of this category include smart appliances (TVs, thermostats, home speakers, etc.) that connect over a Wi-Fi router with OptOutCode on, and IoTs that connect to the internet either directly (e.g. an automobile with a telematics connection, a retail store beacon network connected to the retailer's broadband) or indirectly (e.g. an automobile that leverages over Bluetooth the connected smartphone's connection to the internet, a smartwatch that leverages over Bluetooth the connected smartphone's companion app that connects to the internet).
C. For IoTs that collect and locally store unsecured Personal Information and have been connected to a device with OptOutCode on, , OptOutCode should be interpreted as a request from the Consumer to the data Controller (typically the company that owns or ends up owning the device or has otherwise a financial interest in it) to opt out of the Processing of Personal Data for purposes of Targeted Advertising or the Sale of Personal Data pursuant to C.R.S. § 6-1-1306 (1)(a)(I)(A) or (1)(a)(I)(B). The latter includes selling, renting, leasing, or otherwise giving access to the device containing the Consumer data, since that unencrypted and/or unsecured locally stored Consumer personal information is part of the IoT transaction. In other words, when a Consumer turns on OptOutCode, they are objecting to their data being left unprotected on the device. Examples of this category include the devices with OptOutCode on that are being refurbished or recycled (smartphones, routers, laptops, etc.), smart appliances (TVs, thermostats, home speakers, etc.) that connected to devices with OptOutCode on that are being refurbished or recycled, and vehicles that connected to devices with OptOutCode on that are rented, sold, remarketed, or recycled.

We want to remark that we checked that OptOutCode meets all the requirements under the law to be considered a valid Universal Opt-Out Mechanism, not only in Colorado, but across all the States that have a UOOM provision, per the table below:

| Requirement | CA | CO | CT | DE | IN | IA | MN | OR | TN | TX | UT | VA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Be based on clear and unambiguous choices made by consumers, rather than on default settings | YES | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Must be consumer friendly, clearly described, and easy to use by the average consumer | YES | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Must be able verify the consumer's state residency | NO | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Required to inform consumers of the universal opt-out mechanism and inform consumers about the opt-out choices | YES | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Must be able to accurately verify whether a consumer has made a valid opt-out request | NO | NO | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Signal sent by a browser or plugin to a website and business indicating a user preference | YES | YES | YES | YES | N/A | N/A | YES | NO | N/A | YES | N/A | N/A |
| Shall respect the signal over any other user-stated preferences when there is a conflict between the the user's global signal and the user's stated preference through a cookie preference center | YES | NO | YES | YES | N/A | N/A | NO | NO[2] | N/A | NO | N/A | N/A |
| Provide a mechanism for the consumer to selectively consent to a business' sale of the consumer's personal information, or the use or disclosure of the consumer's sensitive personal information, without affecting the consumer's preferences with respect to other businesses or disabling the opt-out preference signal globally | YES | NO | NO | NO | N/A | N/A | NO | NO | N/A | NO | N/A | N/A |
| Cannot unfairly disadvantage the controller or another business | YES | YES | YES | YES | N/A | N/A | YES | NO | N/A | YES | N/A | N/A |

This was not sufficient for us: we also wanted to make sure that OptOutCode met the additional following criteria:

1. OptOutCode is backwards-compatible with essentially all personal devices.
2. OptOutCode leverages existing, long standing, and future proof communications protocols, making it easy for business to adopt it and resilient and flexible across platforms and future tech developments.
3. The State of Colorado (or anybody else) will not be burdened with having to build, maintain, query, or secure a central database of opting-out users, devices, or apps.
4. Consumers can decide to turn on OptOutCode on their devices today, without needing any new or special software, protocol, or industry standard.
5. The OptOutCode signal is usually sent after contracts are signed and other "consent" is given, hence overrides default broad consents that are common practice.
6. Companies' IoTs/Apps can detect UOOM and ask Consumers to override its opt-out. That requires companies to get affirmative and explicit consent ("notice and consent" whose terms are buried in legal documents is no longer sufficient).
7. Turning OptOutCode on and off can be automated or semi-automated with code.
8. Additional consumer protections and business processes can build upon OptOutCode (e.g., enable automated DSAR placement and response).

## Please provide the specification for the UOOM.

The spec is extremely simple:

A. The Consumer renames their device by adding "0$S" as the first three letters in the device name.
B. Businesses read the name of the device using established IT protocols, determine if the name starts with "0$S" and, if so, consider it an opt-out.

Companies can easily create their own code to read the first three letters of a name of a device and trigger the opt-out request, but for illustrative purposes Privacy4Cars is including examples of code and sample technical implementations across a variety of platforms below. We have also added sample code on how to automate (for Android), or semi-automate (for iOS) the renaming of a smartphone to turn on or off OptOutCode with a simple "switch" on an app. If, as we hope, the Colorado Department of Law decides to shortlist OptOutCode for consideration of a valid UOOM, Privacy4Cars plans to release a simple and free app for Consumers to be able to turn on and off OptOutCode (and test if OptOutCode is on or off).

## Android

Android code to enable renaming a device with the □$S prefix (code also checks if "□$S" was already set as a prefix). Please note, this automation will require permissions.

```
/**
 * "Ensure that you check Android permissions before executing read/write operations with the
Bluetooth Manager."
 *  Additional checks like if name already have prefix o$S then show user information or show another
flow to user could be done
 * <!--Before Android 12 (but still needed location, even if not requested)-->
 * <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
 * <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
 * <uses-permission android:name="android.permission.BLUETOOTH" android:maxSdkVersion="30"
/>
 * <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
android:maxSdkVersion="30" />
 *
 * <!--From Android 12-->
 *  <uses-permission android:name="android.permission.BLUETOOTH_SCAN"
android:usesPermissionFlags="neverForLocation" />
 * <uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
 */
@SuppressLint("MissingPermission")
fun renameBluetoothDeviceName(context: AppCompatActivity) {
    val btManager = context.getSystemService(Context.BLUETOOTH_SERVICE) as BluetoothManager
    val mBluetoothAdapter = btManager.adapter

    if(!isBtNameAlreadyHavePrivacyPrefix(context)) {
        mBluetoothAdapter.name = "o\$S " + mBluetoothAdapter.name
    }else{
        Toast.makeText(context, "Device name already have a prefix o\$S.",
Toast.LENGTH_SHORT).show()
    }
}

/**
 * This method can be used to check if device name already have a o$S prefix or not.
 */
@SuppressLint("MissingPermission")
fun isBtNameAlreadyHavePrivacyPrefix(context: AppCompatActivity): Boolean {
    val btManager = context.getSystemService(Context.BLUETOOTH_SERVICE) as BluetoothManager
    val mBluetoothAdapter = btManager.adapter
    return mBluetoothAdapter.name?.startsWith("o\$S") == true
}
```

Android sample code an app can use to read the name of the device on which it is installed and determine if it includes the "□$S" prefix and trigger an opt-out or not.

```
/**
 * This method can be used to check if device name already have a 0$S prefix or not to determine
 * if UniversalOptOut Enabled or not,This method returns boolean true/false
 * where true mean UniversalOptOut Enabled
 * and false mean UniversalOptOut Disabled
 */
@SuppressLint("MissingPermission")
fun isUniversalOptOutEnabled(context: AppCompatActivity): Boolean {
    val btManager = context.getSystemService(Context.BLUETOOTH_SERVICE) as BluetoothManager
    val mBluetoothAdapter = btManager.adapter
    val isOptedOut = mBluetoothAdapter.name?.startsWith("0\$S") == true
    if (isOptedOut){
        Log.e("UniversalOptOutStatus","UniversalOptOut Enabled")
    }else{
        Log.e("UniversalOptOutStatus","UniversalOptOut Disabled")
    }
    return isOptedOut
}
```

Android sample code an app can use to read the name of a device the smartphone is paired with over Bluetooth (in case the "□$S" prefix is applied not to the smartphone but to a device that is set with a universal opt-out signal).

```
/**
 * Following function can be used to retrieve device name if it is already connected to a Bluetooth device.
 * If required connection update as soon as they happen i.e., Listening to connected or disconnected state
 * following broadcast receivers can be used
 * BluetoothDevice.ACTION_ACL_CONNECTED
 * BluetoothDevice.ACTION_ACL_DISCONNECT_REQUESTED
 * BluetoothDevice.ACTION_ACL_DISCONNECTED
 */
@SuppressLint("MissingPermission")
fun getConnectedDeviceName(context: AppCompatActivity): String? {
    val btManager = context.getSystemService(Context.BLUETOOTH_SERVICE) as BluetoothManager
    for (device in btManager.adapter.bondedDevices) {
        if (isConnected((device))) {
            return device.name
        }
    }
    return "No device connected currently."
}

private fun isConnected(device: BluetoothDevice): Boolean {
    return try {
        val m: Method = device.javaClass.getMethod("isConnected")
        m.invoke(device) as Boolean
    } catch (e: Exception) {
        throw IllegalStateException(e)
    }
}
```

## Apple

iOS code to enable renaming a device with the □$S prefix. Please note, this automation will require permissions.

```
@IBAction func changeNamePressed(_ sender: Any) {

    let alert = UIAlertController(title: "Alert", message: "Go to About > Name and Type \"o$S\" in front of your current device name to enable the o$S Global Opt-Out Mechanism", preferredStyle: .alert)

    let cancelAction = UIAlertAction(title: "Cancel", style: .destructive)

    let okAction = UIAlertAction(title: "Ok", style: .default, handler: { action in

      self.openSettings()

    })

    alert.addAction(cancelAction)

    alert.addAction(okAction)

    self.present(alert, animated: true)

  }

  func openSettings() {

    if let url = URL(string:UIApplication.openSettingsURLString) {

      if UIApplication.shared.canOpenURL(url) {

        UIApplication.shared.open(url, options: [:], completionHandler: nil)

      }

    }

  }
```

iOS sample code an app can use to read the name of the device on which it is installed and determine if it includes the "□$S" prefix and trigger an opt-out or not.

```
if UIDevice.current.name.hasPrefix("o$S") {

      print("Universal Opt Out Enabled")

    } else {

      print("Universal Opt Out Disabled")

    }
```

## Windows

The Windows OS applications (e.g., Chrome browser) are built using ASP.NET framework (C# programming language). This framework provides four ways to get the name of the machine or computer:

1. string MachineName 1 = Environment.MachineName;
2. string MachineName2 = System.Net.Dns.GetHostName();
3. string MachineName3 = Request.ServerVariables[“REMOTE_HOST”].ToString(); and
4. string MachineName4 = System.Environment.GetEnvironmentVariable(“COMPUTERNAME”).

OS sample code

```
// Sample for the Environment.MachineName property

using System

class Sample

{

   public static void Main()

   {

   Console.WriteLine();

//  <-- Keep this information secure! -->

   Console.WriteLine("MachineName: {0}", Environment.MachineName);

   }

}

/*
```

## Bluetooth

Bluetooth Classic mechanism to read a device name:

A. When a device (e.g., a smartphone) is in discoverable mode, the second device (e.g., the infotainment system of a vehicle) can read the first device’s name with an Extended Inquiry; or
B. When devices are paired, the commands LMP_NAME and LMP_NAME_RES are used at the beginning of the session (each device can read the other device’s name).

Bluetooth Low Energy (BLE) mechanism to read a device name:

A. When smartphone (Central) is connectable mode, BLE device (Peripheral) can read Central's Device Name via GATT (typically not protected, pairing not required); or
B. When Central and Peripheral are paired, each device can read the other device's name at the beginning of the session via the command GATT READ CHARACTERISTIC.

## What steps would a Consumer have to take to use the UOOM? Please include whether the UOOM will be a default setting for a tool that comes pre-installed with a device.

As explained above, all a Consumer needs to do to turn OptOutCode on is rename their device by adding "0$S" as the first three letters in its name.

This is not a default setting, and the act of renaming the device (e.g., their smartphone, their laptop, their Wi-Fi router) should be considered "a Consumer's affirmative, freely given, and unambiguous choice to opt out of the Processing of Personal Data".

## What steps would a Controller have to take to detect the UOOM?

The Controller will have to query the name of the device and determine if the first three letters are "0$S". We provided above a number of examples on how this can be accomplished across a variety of platforms and use cases.

## When your UOOM is used by a Consumer, how can a Controller determine that the Consumer using the UOOM is a Resident of the State of Colorado and that the use represents a legitimate request to opt out of the Processing of Personal Data?

Controllers have many alternative and possibly redundant methods to make that determination, depending on the specific situation and use case. By means of illustration, here are some simple examples on how Controllers may make that determinantion.

For applications and software running on devices with OptOutCode on, the data Controller could determine if the user is a Colorado resident in multiple ways. For instance, as GPC does, it could use the IP address of the device's connection. Often users of applications and programs must enter a licensing agreement/registration process, so the company would have access to the state of residency from their own records. In some online services (e.g., social media platforms) it is normal for Consumers to volunteer information about where they reside even when not mandatory. Companies could also use identity services to, for instance, tie a unique identifier of the device to the owner and their residency.

For IoTs, in addition to the methods described above, Controllers would typically know the location of where the IoT is deployed. For instance, a Bluetooth sniffer installed in a retail store to track shoppers in Colorado would have to assume that those shoppers are Colorado residents.

In other cases, a financial transaction related to an IoT could be informing Controllers of the residency of Consumers. For instance, a rental car company who is shown a Colorado driver license would know the vehicle is given in use to a Colorado resident hence would need to respect the UOOM. A retailer selling an IoT (e.g., a smart refrigerator, a smartTV) may know from a loyalty program card, a credit card transaction, or some other registration (e.g., for warranty and support). In some cases, IoT transactions may require showing an ID or otherwise register the transaction with the State (as in the case for instance with vehicles leased or purchased at dealerships).

## What are the costs of use, implementation, and detection of the UOOM by Consumers and Controllers?

Since OptOutCode leverages an existing feature (the ability to name devices), it is completely free for Consumers, and it takes only a few seconds for a Consumer to activate the opt-out signal by renaming the device to include the "□$S" prefix. Arguably, OptOutCode is the cheapest, simplest, and fastest possible UOOM standard for Consumers to adopt with the highest potential impact on Consumer privacy beyond what is currently allowed by GPC.

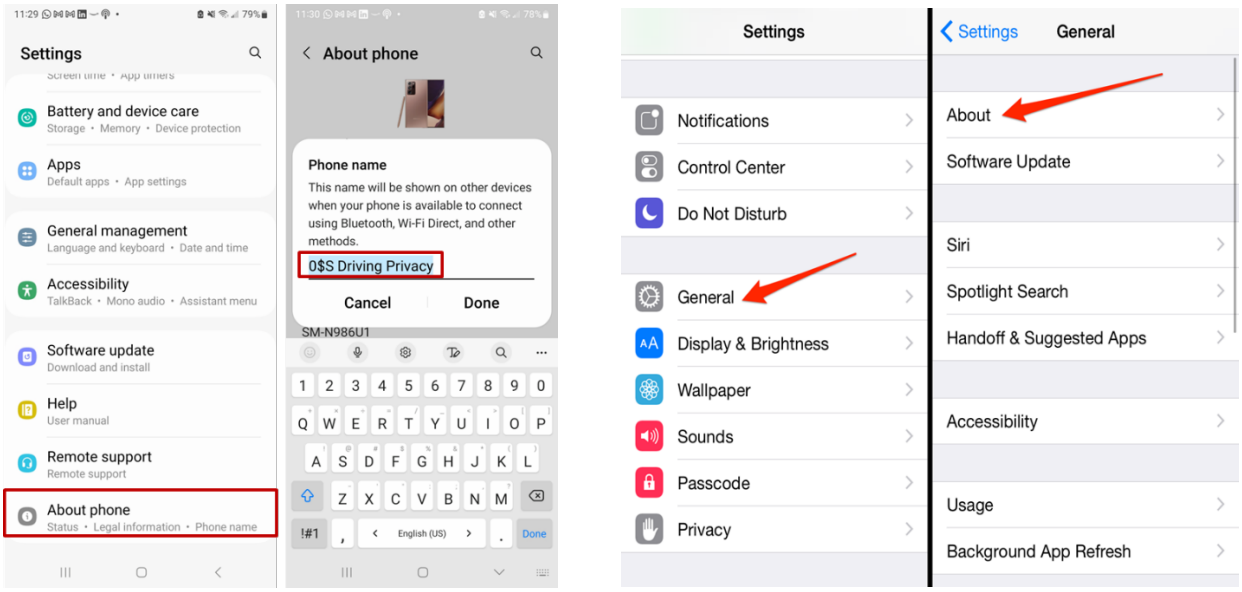


*Illustration: how to change the name of an Android and Apple phone to turn on OptOutCode*

All the technology required by Controllers to listen to this device is also available and free because they would be leveraging preexisting protocols to read the names of the devices. As we pointed out in the preface, Google has a somewhat similar protocol to allow users to opt-out of their Wi-Fi routers' signals[5] to be used by Google Maps, providing an independent industry validation that a portion of the name of a device can be used as an opt-out signal. Adding and rolling-out the code needed to be able to listen to this signal is trivial and a pure exercise of will, not of skill.

Of course, once they listen to the signal, companies would have to make changes on how to process the opt-out preference that those Consumers are expressing. Arguably, since Controllers have to be able to respect opt-out requests from Consumers anyway, if they have already implemented a process to automate processing of opt-outs the marginal cost companies would face to roll-out those code and process changes should be zero or negligible.

Companies that are in the business of renting/selling/leasing/sharing or otherwise give common access to IoTs that store unencrypted/unsecured personal information of Consumers would have to

---

[5] https://www.androidpolice.com/2020/05/03/opt-out-wi-fi-from-google-location-services/. See also, https://support.google.com/maps/answer/1725632?hl=en

implement a data destruction program in line with the NIST[6], EPA[7], international industry standard bodies[8]. Since those standards are already broadly adopted for consumer electronics, or arguably should have already been put in place to safeguard Consumers, also in this case the marginal cost is null or negligible.

Arguably, OptOutCode is the cheapest possible UOOM standard for companies to comply to.

## Is the UOOM based on an open system or standard that is free for adoption without permission or on fair, reasonable, and non-discriminatory terms?

Since OptOutCode leverages an existing feature (the ability to name devices), it is completely free for both Consumers and companies to use.

As the inventor, Privacy4Cars is only asking that if consumer-facing applications are built that leverage this UOOM standard (e.g., to automate turning on/off the UOOM):

A. The name "OptOutCode" and/or logo of OptOutCode should be prominently displayed in the context of the feature (logos are available for download on the website https://OptOutCode.com); and
B. The tagline "by Privacy4Cars®" or "A Privacy4Cars® Universal Opt-Out Concept" (with the word Privacy4Cars being a hyperlink to https://privacy4cars.com) should be prominently displayed in the context of the feature.

## How will personal data collected in connection with the Consumer's utilization of the UOOM be used, disclosed, or retained? Please include whether the UOOM will be used as part of a digital fingerprint.

One of the best features of OptOutCode is that it does not require to build, maintain, query, or secure a central database of opting-out users, devices, or apps. The signal is detected locally: either by the programs and apps running on the device that has OptOutCode turned on (e.g., an app on a smartphone), or by other devices that connect to the device of the Consumer that has OptOutCode turned on (e.g., a SmartTV connected to a Wi-Fi router with OptOutCode on, a fitness watch connected to a smartphone with OptOutCode on). This architecture makes it privacy-preserving, more secure, without a single point of failure, cheaper, and faster. If fact, Controllers may not need to maintain a database of OptOutCode users at all, since the '0$S" prefix can be read every time and trigger action (or not) with every instance. We also want to point out that device names are not unique identifiers of devices (as opposed to, for instance, the Bluetooth MAC address of a device or the IMEI of a phone) and they are broadcast anyway, consequently adding the OptOutCode would not be effective in creating a digital fingerprint.

## Has the UOOM been adopted by Consumers or Controllers?

At Privacy4Cars we have done extensive testing over the last two years proving the "□$S" prefix detection would work in various situations and scenarios. We plan to release an application and a

---

[6] https://csrc.nist.gov/pubs/sp/800/213/final

[7] https://www.epa.gov/smm-electronics/certified-electronics-recyclers

[8] https://e-stewards.org/learn-more/for-enterprises/overview/whats-the-e-stewards-standard/

Software Development Kit ("SDK") to make it easier for developers to adopt OptOutCode as a feature inside their own applications in early 2024.

As pointed out before, Consumers can turn on, for free and in seconds, the OptOutCode on their devices. Companies can use widely known and established techniques to detect it.

As pointed out, one tech giant has already adopted a similar renaming mechanism to allow users to opt their access point out of location services. Google Map users[9] can opt their access point out of Google Location services by changing the SSID (name) of their Wi-Fi access point (their wireless network name) so that it ends with "_nomap". For example, if their SSID is "12345", you would change it to "12345_nomap". If Google can accept the "_nomap" suffix, all other Controllers should be able to accept the "0$S" prefix.

The only two hurdles to the adoption of OptOutCode as a broadly accepted UOOM standard are:

A. Whether the Colorado Department of Law will agree to consider OptOutCode as a valid UOOM standard; and
B. Raising awareness among Consumers. We believe that if the Colorado Department of Law will agree to consider OptOutCode as a valid UOOM standard, the resulting media coverage and the outreach of various Consumer organizations would help achieve the necessary level of awareness to drive substantial Consumer adoption.

## Has the UOOM been approved or is being actively considered by a widely recognized, legitimate standards body after multistakeholder participation in the standards-making process? If so, which?

Since OptOutCode leverages an existing feature set (managing device names), the underlying standards for naming, reading, transmitting, and otherwise processing device names has long been established by a variety of industry standards organizations, including the Bluetooth SIG and IEEE Standards Association.

OptOutCode also has the support of the Electronic Frontier Foundation (EFF), Electronic Privacy Information Center (EPIC), and the Surveillance Technology Oversight Project (S.T.O.P.).

## Which individuals and organizations have been involved in developing the UOOM?

OptOutCode is an original idea by Andrea Amico, founder & CEO of Privacy4Cars. Development and testing was led by Privacy4Cars. In 2023, we reached out to a number of associations, experts from both academia and industry, and government officials asking for feedback and suggestions as we started to develop the technical documentation and syndicate the concept. We already gained the official support of the Electronic Frontier Foundation (EFF), Electronic Privacy Information Center (EPIC), and the Surveillance Technology Oversight Project (S.T.O.P.). We hope and expect that more will join in supporting this idea.

---

[9] https://support.google.com/maps/answer/1725632#zippy=%2Chow-do-i-opt-my-access-point-out-of-google-location-services

## How did you solicit and consider stakeholder input while developing the UOOM and/or this application?

In 2023, we reached out to a number of associations, experts from both academia and industry, and government officials asking for feedback and suggestions as we started to develop the technical documentation and syndicate the concept. We already gained the official support of the Electronic Frontier Foundation (EFF), Electronic Privacy Information Center (EPIC), and the Surveillance Technology Oversight Project (S.T.O.P.). We are not naming the many others who kindly spared their time and expertise to help us prepare for this application but have chosen to remain anonymous.

## Which stakeholders provided input on the UOOM?

In 2023, we reached out to a number of associations, experts from both academia and industry, and government officials asking for feedback and suggestions as we started to develop the technical documentation and syndicate the concept. We already gained the official support of the Electronic Frontier Foundation (EFF), Electronic Privacy Information Center (EPIC), and the Surveillance Technology Oversight Project (S.T.O.P.). We are not naming the many others who kindly spared their time and expertise to help us prepare for this application but have chosen to remain anonymous.

## Is the UOOM likely to comply with the requirements of other jurisdictions that recognize universal opt-out mechanisms or signals by law?

At the time of this writing, we are aware of seven U.S. state privacy laws that include UOOM provisions:

(a) California Consumer Privacy Act § 1798.185 is effective now;
(a) Colorado Privacy Act, 4 CCR 904-3, Rules 5.03 - 5.07 becomes effective on 7/1/2024;
(b) Texas Data Protection Act § 541.055(e) becomes effective on 7/1/2024;
(c) Connecticut Data Privacy Act § 6(A)(ii) becomes effective on 1/1/2025;
(d) Delaware Privacy Act § 12D-105 becomes effective on 1/1/2025;
(e) Montana Consumer Data Privacy Act § 6 becomes effective on 1/1/2025; and
(f) Oregon Consumer Privacy Act § 12 becomes effective on 1/1/2026.

OptOutCode meets all the requirements to be a valid UOOM under all the privacy state laws:

| Requirement | CA | CO | CT | DE | IN | IA | MN | OR | TN | TX | UT | VA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Be based on clear and unambiguous choices made by consumers, rather than on default settings | YES | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Must be consumer friendly, clearly described, and easy to use by the average consumer | YES | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Must be able verify the consumer's state residency | NO | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Required to inform consumers of the universal opt-out mechanism and inform consumers about the opt-out choices | YES | YES | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Must be able to accurately verify whether a consumer has made a valid opt-out request | NO | NO | YES | YES | N/A | N/A | YES | YES | N/A | YES | N/A | N/A |
| Signal sent by a browser or plugin to a website and business indicating a user preference | YES | YES | YES | YES | N/A | N/A | YES | NO | N/A | YES | N/A | N/A |
| Shall respect the signal over any other user-stated preferences when there is a conflict between the the user's global signal and the user's stated preference through a cookie preference center | YES | NO | YES | YES | N/A | N/A | NO | NO[2] | N/A | NO | N/A | N/A |
| Provide a mechanism for the consumer to selectively consent to a business' sale of the consumer's personal information, or the use or disclosure of the consumer's sensitive personal information, without affecting the consumer's preferences with respect to other businesses or disabling the opt-out preference signal globally | YES | NO | NO | NO | N/A | N/A | NO | NO | N/A | NO | N/A | N/A |
| Cannot unfairly disadvantage the controller or another business | YES | YES | YES | YES | N/A | N/A | YES | NO | N/A | YES | N/A | N/A |

## Has the UOOM already been recognized by any jurisdiction?

This application in Colorado is the first official request for any state to support OptOutCode as a valid UOOM. We plan to send letters undersigned by the various supporters of OptOutCode to the following agencies and offices asking them to also consider OptOutCode as a valid UOOM:

State Level:

1. California Attorney General Office
2. California Privacy Protection Agency
3. Colorado Attorney General Office
4. Connecticut Attorney General Office
5. Delaware Attorney General Office
6. Montana Attorney General Office
7. Oregon Attorney General Office
8. Texas Attorney General Office

Federal Level:

1. Federal Communications Commission
2. Federal Trade Commission
3. U.S. Department of Commerce
4. U.S. House Energy and Commerce Committee – Consumer Protection
5. U.S. Senate Energy and Commerce Committee – Consumer Protection

Foreign countries:

1. Canada: Office of the Privacy Commissioner of Canada
2. EU: European Data Protection Agency
3. EU: European Data Protection Board
4. UK: Information Commissioners' Office

## Is the UOOM flexible, especially in light of changing user preferences, business model changes, and shifts in the laws of other regulations?

Yes. OptOutCode is extremely flexible and resilient to technology and business model changes because it relies on preexisting long-standing and future-proof mechanisms and protocols on how IoTs and apps work.

## What is the scope of the UOOM? a. Does the UOOM function with phones, computers, and/or any other devices? Does the UOOM provide just one opt out right as opposed to both?

OptOutCode is the first truly "*Universal*" opt-out mechanism because it is compatible with smartphones, laptops, tablets, routers, the apps that run on them and the IoTs they connect to, including vehicles, smart appliances, tracking beacons, and more.

According to 4 CCR 904-3, Rule 5.02(C):

> *A Universal Opt-Out Mechanism may:*
>
> 1. *Express a Consumer's choice to opt out of the Processing of Personal Data for both the Processing of Personal Data for purposes of Targeted Advertising and Sale of Personal Data; or*
> 2. *Express a Consumer's choice to opt out of the Processing of Personal Data for only one specific purpose, either Targeted Advertising or Sale of Personal Data alone.*

At this time, for simplicity, when a Controller detects OptOutCode's "0$S" prefix in the name of a device, it should consider it a Consumer's choice to opt out of both the Processing of Personal Data for purposes of Targeted Advertising and Sale of Personal Data. In an earlier section of this document we illustrated how we believe OptOutCode should be interpreted in various use cases (mainly to take into account certain peculiarities of IoTs). In an earlier section we also suggested how Controllers may obtain consent from Consumers to override the OptOutCode opt-out request.

That said, it is conceivable that OptOutCode could be later enhanced to offer a "menu" of actions that would be triggered by the signal, to allow for more options and rights granted to Consumers. This could be for instance easily accomplished by adding additional characters after the first three "0$S". Such a menu could be published and made publicly available in a variety of places, including on https://optoutcode.com so to avoid any ambiguity on how the signal should be interpreted.

## Has the UOOM been vetted by expert reviewers? Has it been tested in laboratory or real-world environments? If yes, please share the results of any tests or reviews.

As stated before, Privacy4Cars has been testing OptOutCode for the past two years and proved that the signal can be easily set (including through code, samples published above) and detected across a variety of devices, platforms, and use cases. We have also pointed out that Google has an internal standard based on modifying the name of devices as a way for Consumers to communicate an opt-out. We are leveraging existing and long-standing standards that can be independently verified.

We have had several experts and advocates review our work (including people who were involved with the development and roll-out of GPC). Some of them have already agreed to officially support the adoption of this UOOM standard.

## Are there any additional factors that you would like the Attorney General to consider in reviewing your UOOM application?

In closing, we want to append the letter co-signed by the organizations that support OptOutCode as a valid UOOM.

November 6, 2023

Office of the Attorney General
Colorado Department of Law
Attn: Attorney General Phil Weiser
1300 Broadway, 10th Floor
Denver, CO 80203
Email: ag@coag.gov

**Re: Letter of support for "OptOutCode": a new Universal Opt-Out Mechanism compatible with smartphones, laptops, tablets, routers, the apps that run on them and the IoTs they connect to, including vehicles, smart appliances, tracking beacons, and more.**

We commend the Colorado State Attorney General Office for its role in protecting consumers and enforcing the provisions of the Colorado Privacy Act (CPA). An important provision is the obligation of businesses to honor and comply with Universal Opt-Out Mechanisms.

Global Privacy Control ("GPC") has successfully established itself as a Universal Opt-Out Mechanism standard for online browsing which has both benefited consumers who are no longer facing "cookie walls" and consent fatigue as well as businesses with an online presence who through standardization have made compliance simpler and cheaper. Today, however, a greater and greater amount of data collection happens outside of interactions through a browser. In particular, interactions with IoTs and through applications remain without an accepted Universal Opt-Out Mechanism standard.

American consumers spend an average of 4-5 hours a day in Apps. In addition, American households have 22 connected devices in average. The number of consumer and industrial IoTs is expected to more than double in the next 4 years to 16.7 billion devices globally (about a third in North America). As a consequence, consumers are increasingly surveilled through connected technologies for which a Universal Opt-Out Mechanism does not exist… until today!

Privacy4Cars developed a new Universal Opt-Out Mechanism called "OptOutCode". It simply requires a device owner to modify the name of a device by adding the standardized prefix "0$S". For example, the owner of a smartphone would opt-out of the sale or sharing of their personal data collected by their phone, by the apps running on their phone, and by the IoTs connected to their phone by simply turning OptOutCode on their phone, e.g., by changing the name of their phone from "My Phone" to "0$S My Phone."

Originally intended to help consumers opt-out from certain data processing in vehicles, it became apparent that the same protocol could be adopted to most categories of computing devices as well. The standard has many advantages, giving it the potential – if it becomes accepted as a valid Universal Opt-Out Mechanism under the law – to be immediately scaled and adopted by consumers and businesses alike, including:

1. **It is truly Universal:** OptOutCode is compatible with smartphones, laptops, tablets, routers, the apps that run on them and the IoTs they connect to, including vehicles, smart appliances, tracking beacons, and more.
2. **It's fully decentralized:** OptOutCode does not require to build, maintain, query, or secure a central database of opting-out users, devices, or apps. The signal is detected locally: either by the programs and apps running on the device that has OptOutCode turned on (e.g., an app on a smartphone), or by other devices that connect to the device of the Consumer that has OptOutCode turned on (e.g. a SmartTV connected to a Wi-Fi router with OptOutCode on, a fitness watch connected to a smartphone with OptOutCode on). This architecture makes it privacy-preserving, more secure, without a single point of failure, cheaper, and faster.

3. **It's Consumer-friendly:** All Consumers need to do to turn on OptOutCode is simply rename their devices by adding "0$S" as the first three characters. For instance, rename their phone from "My Phone" to "0$S My Phone". Most Consumers can do it themselves by going through the settings of their devices in less than a minute, but we have also written code to automate the task of turning OptOutCode on or off with a simple "switch" on an app.
4. **It's business-friendly:** Businesses can easily read and parse the "0$S" opt-out code from each device using backward-compatible and future proof protocols that require no special authorizations. We want to thank Professor Sebastian Zimmeck of Wesleyan University (credited with inventing GPC) for recently pointing out to us that Google has a somewhat similar protocol to allow users to opt-out of their Wi-Fi routers' signals[1] to be used by Google Maps, providing an independent industry validation that a portion of the name of a device can be used as an opt-out signal. If Google can accept the "_nomap" suffix to opt users out, then other businesses or Controllers should be able to accept and honor the "0$S" prefix.

We are asking the Colorado State Attorney General's Office and Colorado Department of Law to support the proposed OptOutCode Universal Opt-Out Mechanism by approving it as a valid consumer request to:

(a) Not sell their personal data collected by Apps and/or Connected Devices/IoTs and limiting the sharing of their personal data to other companies;
(b) Being removed from marketing, sale, and other communications (both operated by the company, its affiliates, and its third parties including service providers; and
(c) Having any personal data collected from them and stored on the device/IoT cleared/deleted to prevent it from being exposed to other individuals.

The undersigned have reviewed the technical documentation and reasoning behind the "0$S" protocol for Universal Opt-Out, and we support its adoption.

We remain willing and eager to work with the Colorado State Attorney General's Office and Colorado Department of Law to expand the current Universal Opt-Out protections consumers enjoy on their browsers through GPC to interactions they have though the apps on their phones or computers or with devices they connect to or IoTs that track them through OptOutCode. If you have any questions, please don't hesitate to reach out to Andrea Amico, inventor of OptOutCode and Privacy4Cars founder and CEO, at andrea@privacy4cars.com.

Signed,

Privacy4Cars, Inc.

Electronic Frontier Foundation (EFF)

Electronic Privacy Information Center (EPIC)

Surveillance Technology Oversight Project (S.T.O.P.)

---

[1] https://www.androidpolice.com/2020/05/03/opt-out-wi-fi-from-google-location-services/. See also, https://support.google.com/maps/answer/1725632?hl=en.