Technical Disclosure Commons

Defensive Publications Series

February 09, 2019

BEHAVIORAL RANDOM NUMBER GENERATOR IN CRYPTOGRAPHY

David Maluf

Raghuram Sudhaakar

Pradeep Kathail

BEHAVIORAL RANDOM NUMBER GENERATOR IN CRYPTOGRAPHY

AUTHORS:
David Maluf
Raghuram Sudhaakar
Pradeep Kathail

ABSTRACT

Techniques are described herein for using vehicle data for the creation of a seed number for cryptography generation in-car as a potential replacement for private keys. In this approach, each vehicle generates its keys by itself and is also able to rekey independently without the need for a central key generation system. Unlike vehicle keys today, which are generated centrally (mostly at manufacture time) and thus are exposed to a single source of failure if the key generator is compromised, the impact of a key compromise as described herein is isolated to one vehicle. Thus, whereas a threat today would compromising an entire fleet (e.g., millions) of vehicles, the threat of compromising an entire fleet is significantly minimized using techniques described herein as the scale of the problem now changes from attacking a single point of failure to having to break into multiple vehicles at the same time. With a tunable key refresh rate and independent rekeying, the scale of the problem is made many orders of magnitude more difficult for the attacker.

DETAILED DESCRIPTION

The automotive industry lacks adequate solutions for onboard hardware and software to operate high end cryptography due to cost. Existing systems exhibit weak security systems avoiding the state of the art such as hardware security modules which are expensive high-end encryption systems to be developed in commercial vehicles. True Random Number Generators (TRNGs) have been used as a solution, but TRNG are also expensive to have in vehicles.

Accordingly, techniques described herein use vehicle behavior data to generate seed numbers that will serve as the basis for cryptographic operations in the vehicle.

A vehicle consists of hundreds of sensors and actuators sharing data over a message bus. They are continuously sampling and transmitting the digitized values of physical quantities such as speed, engine Revolutions per Minute (RPMs), acceleration, G-forces, etc. The physical values are determined by the operation of the vehicle. Analysis of these values over time shows that they can be modeled as non-stationary random variables. In other words, these values, when sampled over time, exhibit properties of randomness. In real world scenarios, a user may take a different route to work, drive in different lanes, hit potholes on one day and not on another, be impeded by traffic on one day while have a free ride the next, etc. These factors which are the behavior of the user and/or the vehicle contribute to the measured physical quantities exhibiting randomness.

The Behavioral Random Number Generator (BRNG) method described herein uses these behavioral parameters as the entropy source to generate the seed or random number.

The Controller Area Network (CAN) data may include a combination of various physical quantities and Global Positioning System (GPS) data as well. Additional data may be added to include multimedia data as part of the entropy source. A function is defined that transforms the sample of the entropy source into a seed usable for further cryptographic functions.

Due to the fact that the entropy source is available always and for free, there is no additional cost to the system to maintain an entropy source. Further, the ready availability of the entropy source enables faster rekeying. The system designer now has the ability to balance the strength of the cryptographic key with the frequency of rekeying and is thereby able to further reduce the computational costs of rekeying.

An example of continuously and recursively signing the data could be accomplished cheaply with an MD5 hashing. As such, at every moment a new number could be instantly generated for encryption to generate private and public keys. The BRNG method described herein is not superior to TRNG, but BRNG enables real-time random number generations literally at no or extremely low cost to implement.

The source of entropy may be derived from the actual behavior of the vehicle. The randomness of this entropy source when combined with the ability to rekey at a high frequency and low computational cost leads to an increase in overall strength of the cryptography.

One example may involve recycling through the reset mode of regeneration. On a more detailed level, the process assesses the shortfall of the new number and recomputes a new pair of keys as often as needed. The simplest possibility is to rekey every time the vehicle starts. With more control, private and public key pairs are generated at a random time within a lower and upper window. For example, a new pair of keys are generated periodically (e.g., every 1-2 hours). The frequency of rekeying can be very high and still be efficient as the seed number is always ready. Once a new public key is created, the public key is shared with the list of authorized peers and the public key is uploaded. A vehicle therefore maintains control of the keys at all times.

The net result is that each vehicle has a continuous distinct public key to be shared for use. The choice of the window of time before a key is reset depends on the assumption of how much longer it takes to solve for a key on a preset choice of compute power. These numbers are known. The assumption is that computers take some time to solve for a key.

The uniqueness of the behavior of the seed number generation and the frequency of rekeying isolate attacks to individual vehicles. A threat would need to solve for the private key for every car in a narrow window of time before the key is rebuilt anew. The benefit over the state of the art is security failure isolation.

While BRNG is less efficient than TRNG with respect to absolute entropy, BRNG localization requires the attacker to know the state of the CAN bus system and all the physical values measured to recreate the key. This is only possible if the attacker has full access to the CAN bus. In this case, the cryptography becomes useless as the attacker has direct access to the data.

In one example, the role of the rekeying process may be reversed to the edge instead of a controller. That is, the edge controls private/public key generation and not the controller rekeying the edge. At large scale, the inherent weakness of the BRNG for one edge is taken over by the complexity of independency of the plurality of independent keys. For example, if one vehicle key is broken, the damage is limited. Existing systems have all keys in the controller and if the controller is broken into, all the edge devices are immediately vulnerable.

Methods are described to use a BRNG mechanism in an in-vehicle network to minimize cost and complexity of the Electronic Control Unit (ECU) while enhancing security.

In summary, techniques are described herein for using vehicle data for the creation of a seed number for cryptography generation in-car as a potential replacement for private keys. In this approach, each vehicle generates its keys by itself and is also able to rekey independently without the need for a central key generation system. Unlike vehicle keys today, which are generated centrally (mostly at manufacture time) and thus are exposed to a single source of failure if the key generator is compromised, the impact of a key compromise as described herein is isolated to one vehicle. Thus, whereas a threat today would compromising an entire fleet (e.g., millions) of vehicles, the threat of compromising an entire fleet is significantly minimized using techniques described herein as the scale of the problem now changes from attacking a single point of failure to having to break into multiple vehicles at the same time. With a tunable key refresh rate and independent rekeying, the scale of the problem is made many orders of magnitude more difficult for the attacker.