Securing CAN Bus Communication: An Analysis of Cryptographic Approaches

Jennifer Ann Bruton

M.Sc. in Software Engineering & Database Technologies
National University of Ireland, Galway
Discipline of Information Technology
College of Engineering & Informatics
August 2014

Head of IT Discipline:
Dr. Michael Madden

Research Supervisor:
Dr. Michael Schukat
MSCSED THESIS: CAN BUS SECURITY

Contents

1 Introduction ........................................................................... 11
   1.1 Problem Context .......................................................... 11
   1.2 Thesis Statement ........................................................... 12
   1.3 Significance of the Study .................................................. 12
   1.4 Thesis Scope .................................................................. 13
   1.5 Research Approach & Criteria for Success ......................... 15
   1.6 Project Activities ........................................................... 16
   1.7 Report Organization ....................................................... 17

2 Literature Review .................................................................. 18
   2.1 Introduction .................................................................... 18
   2.2 Computer Security ........................................................ 18
      2.2.1 Definition & Importance of Computer Security ............ 18
      2.2.2 Computer security challenges .................................... 19
      2.2.3 Security mechanisms ............................................... 21
   2.3 Embedded Systems Security ............................................ 25
      2.3.1 Importance of Security for Embedded Systems ............ 27
      2.3.2 Embedded Systems Security Challenges ..................... 28
   2.4 Automotive Security ...................................................... 29
      2.4.1 Automotive Embedded Systems ................................. 29
      2.4.2 Automotive Security Issues ....................................... 32
   2.5 The Controller Area Network (CAN) .................................. 38
      2.5.1 Automotive Communication Networks ........................ 38
      2.5.2 The CAN bus ......................................................... 41
   2.6 CAN Security ............................................................... 46
      2.6.1 Security Concerns ................................................... 46
   2.7 Challenges and Goals ..................................................... 46
      2.7.1 Non-cryptographic Security Approaches for CAN ............ 49
      2.7.2 Cryptographic Security Approaches for CAN ................. 51
      2.7.3 Experimental Test-beds ............................................. 59
   2.8 Literature Review Conclusions .......................................... 60
3 Experimental Platform

3.1 BBB Introduction ........................................... 63
3.2 BBB Features ........................................... 65
3.3 BBB Operating System ................................... 67
3.4 BBB Set-up ........................................... 69
3.4.1 Network Connection ................................ 69
3.4.2 Software Environment ................................ 72
3.5 BBB CAN bus Environment ............................ 78
3.5.1 Physical Cape ................................... 78
3.5.2 Cape Installation ................................ 79

4 CAN Bus Communication ........................................... 81

4.1 Introduction ........................................... 81
4.2 CAN Implementation .................................... 81
4.3 CAN Low-Level Communication ....................... 84
4.4 Socket Communications ................................ 90
4.5 SocketCAN ........................................... 92
4.6 Basic Implementation of SocketCAN ................... 95

5 Secure CAN Bus Communication ................................ 100

5.1 Designing Secure CAN Communications ............... 100
5.2 Designing CAN Communications ....................... 103
5.3 Designing AES-Encrypted CAN Communications .......... 107
5.4 Designing RC4-Encrypted CAN Communications .......... 112
5.5 Designing Authenticated CAN Communications .......... 117
5.6 Designing Key Distribution for CAN Communications ..... 123
5.7 Integrating Design Features for Secure CAN Communications . . 128

6 Testing & Results ........................................... 132

6.1 Testing CAN Communications ........................... 132
6.1.1 Testing basic send-receive ........................... 133
6.1.2 Testing Basic Receive-Send ........................... 134
6.1.3 Testing CAN bus Timing ........................... 134
6.2 Testing Block Cipher Encryption for CAN Communications .... 142
6.2.1 Testing Basic AES Send-Receive .................................. 142
6.2.2 Testing CAN bus Timing with AES .................................. 143
6.3 Testing Stream Cipher Encryption for CAN Communications .... 145
   6.3.1 Testing Basic RC4 Send-Receive .................................. 145
   6.3.2 Testing CAN bus Timing with RC4 .................................. 147
6.4 Testing HMAC Authentication for CAN Communications .......... 149
   6.4.1 Testing Basic HMAC Send-Receive .................................. 149
   6.4.2 Testing CAN bus Timing with HMAC Authentication .......... 151
6.5 Testing Out-of-band SSL Communication ............................... 153
6.6 Testing Overall Design for Secure CAN Communications .......... 156
   6.6.1 Testing Basic Overall Send-Receive Behaviour ................. 156
   6.6.2 Testing CAN bus Timing with Combined Authentication -
       Encryption ...................................................... 159

7 Conclusions ........................................................................ 163
  7.1 Introduction .............................................................. 163
  7.2 Meeting the Success Criteria ........................................ 163
  7.3 Answering the Research Question ................................... 164
  7.4 Future Areas for Research & Development ........................ 168

8 References ........................................................................ 170

Appendix .............................................................................. 192
  A. Sockets ................................................................. 192
  B. BenchMark Code ...................................................... 195
  C. AES Code ............................................................. 199
  D. RC4 Code ............................................................. 200
  E. HMAC Code ........................................................... 201
  F. Key Distribution Code .............................................. 202
  G. Overall Design Code ................................................ 205
# List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Activities</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>Testing CAN Transceiver Nodes</td>
<td>87</td>
</tr>
<tr>
<td>3</td>
<td>Timing for Different Iterations</td>
<td>136</td>
</tr>
<tr>
<td>4</td>
<td>Timing for Different Bitrates</td>
<td>137</td>
</tr>
<tr>
<td>5</td>
<td>Timing for Different Bitrates</td>
<td>138</td>
</tr>
<tr>
<td>6</td>
<td>Timing for Candump</td>
<td>138</td>
</tr>
<tr>
<td>7</td>
<td>Testing Different Devices for Timing</td>
<td>139</td>
</tr>
<tr>
<td>8</td>
<td>Testing Different Devices for Timing</td>
<td>141</td>
</tr>
<tr>
<td>9</td>
<td>Testing Separate Platforms for Timing</td>
<td>141</td>
</tr>
<tr>
<td>10</td>
<td>AES Testing on a Single Node</td>
<td>143</td>
</tr>
<tr>
<td>11</td>
<td>Testing AES for Different Nodes on Same Host</td>
<td>145</td>
</tr>
<tr>
<td>12</td>
<td>Testing AES for Nodes on Different Hosts</td>
<td>145</td>
</tr>
<tr>
<td>13</td>
<td>RC4 Testing on a Single Node</td>
<td>149</td>
</tr>
<tr>
<td>14</td>
<td>Testing RC4 for Nodes on Same/Different Hosts</td>
<td>149</td>
</tr>
<tr>
<td>15</td>
<td>HMAC Testing on a Single Node</td>
<td>152</td>
</tr>
<tr>
<td>16</td>
<td>Testing HMAC for Nodes on Same/Different Hosts</td>
<td>153</td>
</tr>
<tr>
<td>17</td>
<td>Authentication-Encryption Testing on a Single Node</td>
<td>159</td>
</tr>
<tr>
<td>18</td>
<td>Effect of timestamp resolution on authentication success</td>
<td>161</td>
</tr>
<tr>
<td>19</td>
<td>Testing Combined Scheme for Nodes on Same/Different Hosts</td>
<td>162</td>
</tr>
</tbody>
</table>
## List of Figures

1. Automotive Components with Communications, 30
2. CAN frame format 44
3. CAN Transceiver Examples, (SN65HVD230, 2011; MCP2551, 2010) 63
4. BBB Features and Layout 65
5. BBB Image Writer 69
6. BBB Serial Connection 70
7. BBB USB Connection 71
8. BBB Ethernet Connection 71
9. BBB Web Server Response 72
10. BBB Nano Editor Example 75
11. BBB Simple Compilation Example 76
12. BBB BoneScript Function Example 76
13. BBB Cloud9 IDE BoneScript Example 77
14. Eclipse IDE for BBB Example 78
15. BBB Tower Tech TT3201 CAN Cape 79
16. Typical CAN Network with Termination, 82
17. singlelinecheck 82
18. singlelinecheck 84
19. singlelinecheck 85
20. singlelinecheck 85
21. singlelinecheck 86
22. CAN Transceiver Test Circuit Single Chip Outputs, 87
23. CAN Transceiver Test Circuit 2 88
24. CAN Bus Signals for Individual Nodes, 88
25. CAN Transceiver Test Circuit 2, Successful Transmission & Reception 89
26. CAN Transceiver Test Circuit 2, Timing 89
27. Client Socket Connection on BBB 92
28. CAN Communication Layers 93
29. Socket-Based CAN Communication 94
30. Single BBB, Two-Node CAN Bus 97
31. Multiple BBBS, 3-Node CAN Bus 98
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>HMAC Authentication on CAN bus with timestamp</td>
</tr>
<tr>
<td>65</td>
<td>Mean Message Times for Different Payloads with/without Authentication</td>
</tr>
<tr>
<td>66</td>
<td>Key Server Application Running</td>
</tr>
<tr>
<td>67</td>
<td>Key Client Application Running</td>
</tr>
<tr>
<td>68</td>
<td>Key Client Application Request Times</td>
</tr>
<tr>
<td>69</td>
<td>Authentication-Encryption on CAN bus</td>
</tr>
<tr>
<td>70</td>
<td>Authentication-Encryption on CAN bus, short message</td>
</tr>
<tr>
<td>71</td>
<td>Authentication-Encryption on CAN bus, counter change</td>
</tr>
<tr>
<td>72</td>
<td>Mean Message Times for Different Payloads with/without Combined Security Measures</td>
</tr>
</tbody>
</table>
Abstract

The security of the Controller Area Network (CAN) within automotive applications is becoming significantly more important for maintaining a safe and private driving experience but the CAN bus itself has no security features. Recent articles have highlighted the potentially devastating consequences that can arise when this network is successfully compromised. Research into the area of embedded systems security, and automotive security in particular, is a relatively nascent area. Most efforts concerned with improving CAN security using software-based cryptographic methods have focussed on message authentication, where the challenge of dealing with a small packet frame is considerable. The aim of this research is to further investigate the effects of using cryptographic approaches for both encryption and authentication to improve the security of CAN bus communications. This thesis presents an experimental platform based on the BeagleBone Black, SocketCAN and OpenSSL library for testing cryptographic methods on a CAN bus. Different cryptographic schemes concerned with AES and RC4 encryption, HMAC message authentication, and authenticated encryption are designed and deployed on this platform. The experimental results collected as part of this research confirm that the security of the CAN bus can be improved using cryptographic techniques, and that there are consequences with respect to message times as a result of applying these security measures. This research also highlights elements of the CAN bus communications context and aspects of the cryptographic designs that suggest directions for future development.
1 Introduction

1.1 Problem Context

In the modern world, the proliferation of computer systems, together with their networking and interconnection, has created a significant dependence by governments, organizations, and individuals on the data stored, processed, and exchanged using these systems; thus, these computer resources and their data/information are valuable assets that need to be protected. Computer security is concerned with providing the necessary safeguards for an information system in order to protect the confidentiality, integrity and availability of its resources, data and services; any additional benefits of using computer technology are only realized if these systems are secure from malicious attacks. Providing sufficient computer security can be challenging due to the fact that system resources possess multiple vulnerabilities that allow that system to become compromised in such a way that it delivers an incorrect output, or that its confidentiality is endangered, or it operates so slowly that it has effectively been rendered unavailable.

Security countermeasures are employed to ideally prevent a security attack from taking place, or to minimize and/or record the effect of such an attack if prevention cannot be accomplished. No single countermeasure can succeed in thwarting all attacks, so the principle of 'defence-in-depth' is often advocated and adopted, where a number of layers of security, using different strategies, are employed. Cryptographic strategies involve the conversion of data into a secret form so that some, or all, of authentication, confidentiality, integrity and non-repudiation requirements can be achieved.

An embedded system is an electronic device that contains a combination of one or more microprocessors and software to carry out a very specific function within a larger system. Embedded systems have gained huge popularity in recent years and, due to their interconnectivity, feature heavily within the 'Internet of Things' vision. As a result of this interconnectivity, they offer great attack opportunities (attack surface) to malicious adversaries and, at the same time, pose great challenges for security provision due to their limited resources, low
The use of electronics in automobiles to provide better vehicular performance, passenger safety, and enhanced entertainment facilities, involves the networking together of embedded micro-controllers known as electronic control units (ECUs); a standard automobile contains approximately 50-100 of this units. For in-vehicle communications, the Controller Area Network (CAN) is the preferred bus when exchanging data as it is reliable and works well in noisy, electromagnetic environments.

1.2 Thesis Statement

This thesis focuses on the aforementioned CAN bus and the software-based cryptographic measures that could potentially improve its security. The research question to be answers by the thesis is: “what effects will cryptographic approaches have on the security of the CAN bus and whether the consequences for (time) performance that arise from any improvements in security will be acceptable?” The thesis statement is, therefore:

***

An investigation of the use of cryptographic approaches to secure the Controller Area Network (CAN) bus for real-time automotive embedded systems communications, where performance requirements and resource constraints are strictly defined.

***

1.3 Significance of the Study

Security measures to achieve the goals of confidentiality, integrity and availability for automotive embedded systems can be lacking (Koscher et al., 2010) but their provision is becoming increasingly important. Automotive systems security poses a development challenge due to the limited resources in terms of memory, storage, computational power and bus capacity available to embedded devices. The native CAN bus itself has no security and, as the bus operates in broadcast mode, all ECUs or nodes connected to the bus receive the
same messages. Historically, up to about a decade ago, the security of the CAN bus was interpreted as its ability to transfer data with appropriate error detection, error signal and self-monitoring; it was deemed unlikely that an attacker could easily access this network.

Unfortunately, this historical view has recently been proven to be incorrect and internal attacks on the in-vehicle network often specifically target the weaknesses of the CAN bus to compromise elements of the CIA security triad. Recent studies have shown that successfully accessing a single node on the network allows the entire vehicular network to be compromised. Havoc has been shown to ensue when the engine, brakes, doors, power steering, instrumentation, electric windows, radio, warning lights, diagnostic unit, driver information console, and air-bags have been accessed by attackers. The ability of a malicious party to compromise these automotive components leads to significant safety, commercial and data privacy concerns. The CAN bus is also gaining popularity in industrial automation environments and the need for safe operation in these settings is also highly important.

The hard real-time requirements for communications to and from the ECUs, and the constraints of the CAN protocol itself, present challenges for improving the security of the CAN bus. This study will investigate software-based cryptographic approaches to address these challenges, and, if successful, will demonstrate improved CAN bus security without incurring unacceptable delays in communications and without the need for additional, costly, hardware resources.

1.4 Thesis Scope

Given the thesis statement above, and the problem domain outlined, this thesis will examine the following areas of embedded systems and CAN bus technology.

- Automotive field bus technology: the physical properties of the CAN bus and the elements of the CAN bus protocol will be examined; some context for the CAN bus will be created by briefly describing alternative vehicular bus technologies.
• Communication protocols: the CAN bus protocol will be used for communications on the CAN bus; other communication protocols may be used for any out-of-band (not using the CAN bus) communications that may be required, e.g. for key distribution.

• Embedded system / real time operating systems: as the nodes connected to the CAN bus are embedded systems that perform in real-time, the communication system will be supported by an operating system, the appropriate characteristics and mechanisms of which will need to be understood and manipulated.

• Open source software for embedded systems: both the embedded operating system and any software to manipulate and measure the CAN bus messages will be open source.

• Cryptographic approaches for improving the security of the CAN bus, in particular, the elements of computer security that will be explored in this thesis are:

  – Encryption - in order to provide message content confidentially, encryption techniques will be employed; due to the hard real-time nature of the CAN bus, the focus for payload encryption will be symmetric encryption.

  – Message authentication - the basic CAN protocol does not have a field to authenticate the sender, therefore message authentication schemes will be investigated.

  – Key distribution - the distribution of keys to only trusted nodes for securing the communications payload will be explored.

  – Testing and measurement methodologies - the design of appropriate tests to confirm or negate the validity of a given cryptographic approach, and the measurement of time- and/or resource-related performance measures will be established.
7 Conclusions

7.1 Introduction

This section of the thesis contains the insights that have been obtained from the test results collected from the experiments conducted in the previous section. It presents conclusions about how these results address the research question “what effects will cryptographic approaches have on the security of the CAN bus and whether the consequences for (time) performance that arise from any improvements in security be acceptable?” presented in the introduction. Limitations of the designs that are exposed by the results are presented, and opportunities for further research and development are outlined.

7.2 Meeting the Success Criteria

There are a number of success criteria for the thesis set out in Section 1.5 and these can be evaluated based on the results and documentation presented in this report.

An extensive literature review is present that highlights the issues surrounding embedded systems security, automotive security and CAN bus security; this review gathers and evaluates the efforts that have already been made to improve the security of CAN bus communications. This review reveals that few published articles include experimental details from a physical CAN bus, and that many schemes focus on one particular aspect of CAN bus security. The cryptographic approaches identified for improving CAN bus security are message authentication and symmetric encryption, although most articles have concentrated on message authentication with symmetric encryption enjoying very limited utilization.

One of the key contributions of this thesis is the implementation of the proposed security designs on an actual CAN bus experimental platform. The low-cost BeagleBone Black with its AM335x microprocessor is shown to be effective at facilitating CAN bus experiments; the number of CAN bus channels available on each BBB can be extended with the use of a dedicated cape. The use of the Linux kernel to support the CAN bus networking system using SocketCAN and can-utils is a relatively new concept that has been exploited
very successfully in this thesis to deliver the experimental framework for testing CAN bus communications. The utilization of the OpenSSL library to provide both symmetric and asymmetric cryptographic approaches as part of the designs for CAN bus security is also demonstrated as effective, although some of the more recent hash functions and lightweight ciphers are not available with this library. The 3 scenarios established for experimental testing allow a thorough investigation of CAN communications, whether they are to and from the same device, to and from devices on the same micro-controller host, or whether they are to and from devices on separate hosts. The challenges presented by a real network are thus exposed in ways that may not be apparent through software simulation and virtual CAN nodes.

The design and development of software to cater for the different experimental treatments identified in the Introduction to answer the research question is documented in Section 5. It can therefore been concluded that experiments have been successfully established to test the CAN bus platform with no security; with two types of symmetric encryption; with message authentication; for out-of-band key distribution; and for a combination of encryption and authentication. Each of these designs allows the collection of data measurements linked to the timing of CAN bus messages from one node to another and the appropriate presentation, analysis and contextualization of this data allows conclusions to be drawn so that the overall research question can be answered, as indicated in the following discussion.

7.3 Answering the Research Question

The different treatments necessary to address the research question are designed and implemented using the established experimental platform and created software in order to produce results that can inform the answer to the research question.

When there are no cryptographic approaches applied, it has been clearly demonstrated in Figure 34 that it is a relatively straightforward task to access the CAN bus using an oscilloscope (or other probe), to capture data from the the CAN bus, and to interpret this data once the CAN frame layout is known
and understood. It has also been shown in Section 4.3 on low-level CAN communications, that a single CAN transceiver can be used to transmit signals onto the CAN bus. It can be surmised, therefore, that any successful cryptographic approaches will improve this situation in some way, i.e. if the data is encrypted it will be harder to read, if the message has authentication, it will be harder to inject external messages.

The effects of any cryptographic approach will be different for each CAN network, depending on the underlying hardware capacity of that network. It has been shown in Section 6.1.3 that a number of factors can influence the outcome of the experiments in this thesis, and therefore the most appropriate way to carry out any further experiments, is to use exactly the same test configuration each time. This does mean that quantitative results obtained from an individual treatment are not very useful, but comparing results for different treatments using the same experimental platform, allows conclusions to be drawn. The experiments performed when there are no cryptographic effects in place establish a benchmark for this comparison, although a hard limit of 1\(ms\) is also defined.

Using symmetric block cipher encryption with a pre-shared key for CAN bus communication is shown to be effective at making the original message payload secret (Figure 55), and the encryption time not very onerous (Figure 57), so it can be concluded that this is an effective way of securing the confidentiality of CAN messages. Unfortunately, when standard CAN frames of 8 bytes are used, a 16-byte block cipher means that two CAN messages are required for each original message, effectively doubling the message time compared to using no cryptographic approach at all. The doubling of the message time for slower CAN controllers means that the overall time may go above 1\(ms\) which is unacceptable; it is therefore concluded that this block cipher method should not be used for standard CAN frames; it should be noted that if the CAN FD standard with a 64-byte data field is used, this may be an appropriate approach but preferably when message payloads are close to a multiple of 16 bytes.

The adoption of a stream cipher eliminates the problem of additional messages caused when using the block cipher; it is demonstrated that this encryption also effectively hides the original message content (Figure 58) but
that it generates the ciphertext on a byte-by-byte basis so that the encrypted payload is exactly the same size as the original. The weaknesses inherent in the RC4 stream cipher are mitigated in some way by the dropping of the first 512 bytes of the keysteam and by adding a changing value (counter) to the encryption key. However, the use of the CAN identifier field for the nonce is not ideal in this scenario (though it might be acceptable when a 29-bit identifier is used) and when the counter resets, a new encryption key should be used. The increase in message trip time is relatively modest (Table 13), therefore it can be concluded that using a stream cipher does improve CAN bus security without a detrimental effect on message times but that additional effort must be employed to manage the uniqueness of the encryption key.

This work has confirmed that message authentication can be achieved on the CAN bus using HMAC (Figure 62); due to the limited length of the standard CAN data field, the HMAC needs to be truncated and a value of 32 bits was chosen for this. The effect of appending the HMAC to the message limits the message to 4 bytes, and messages longer than this will need to be sent over two frames; again, this is not likely to be an issue with CAN FD 64-byte data frames. The additional time required for messages using HMAC is approximately 40%, which is a noticeable increase, but still leaves the time lower than the hard 1ms limit. If multiple frames are required, then this limit will be breached but this is a natural consequence of appending the macTag to a message. It can be concluded, therefore, that a HMAC approach will improve message authentication but for some messages within a limited frame size, the additional time overhead may be undesirable.

The cryptographic schemes presented in this thesis rely on the robustness of their secret keys, and these keys need to be refreshed in order to ensure that their reuse does not expose vulnerabilities to attackers. New CAN protocols have been previously been established by other authors to distribute keys via the CAN bus but these mechanisms have costs associated with them in terms of messages forgone in order to send the keys, they are not standard and they must be implemented outside of the CAN controller which deals with the regular CAN protocol. Distributing keys out-of-band could leverage existing secure network
protocols such as SSL/TLS without impacting the messages on the CAN bus itself. This thesis does demonstrate that this is possible for a considerable time overhead (compared to the CAN bus message times) but has not been integrated with the cryptographic approaches at this point. A conclusion that can be drawn, however, is that cryptographic approaches can improve CAN bus security if an effective key distribution is in place and that the time overhead associated with an out-of-band mechanism suggests that the keys should be requested in advance of when they are actually going to be needed.

One of the main contributions of this thesis is the production of experimental results for a cryptographic scheme that involves both message authentication with timestamping and symmetric encryption. The results obtained clearly show that CAN bus security is improved, and that message times are considerably increased ($\approx 60 - 70\%$) but still lie well below the 1ms threshold. However, there are numerous negative consequences for the particular scheme that has been implemented: the counter for the encryption key nonce is limited to 2 bytes, requiring that the encryption key is changed at least every $2^{16}$ messages; the HMAC tag is only 2 bytes long which is not recommended by NIST and would require that the authentication key is renewed every 2.56s; the message payload is limited to 4 bytes which would require additional frames for longer message content; the timestamp is highly dependent on synchronous time over the network is suspected to be adversely affected by CAN controller buffering. This last point highlights the benefits of using an experimental platform for the testing as the proposed timestamp feature works extremely well when all the nodes are on the same host but when they exist on different hosts, the timestamp is rendered ineffective as a method for adding uniqueness to the authentication scheme. It is possible that this effect could be missed when simulating a network or when using only a single, multi-channel host.

It is considered that the research question has been answered over a series of different experiments and that the outcomes of these experiments can help to inform future research and work.
7.4 Future Areas for Research & Development

One of the main constraints with the CAN bus communications for this thesis was the lack of availability of CAN FD Controllers on the BBB cape. The investigation of cryptographic methods for the very limited size standard CAN frame is interesting, but the effects and consequences of these cryptographic approaches for the larger payload could also be evaluated.

The overall design for authenticated encryption assumes that all messages need to be authenticated and encrypted; it is likely in a real-world scenario that this might not be the case and the overhead associated with the applying the cryptographic approach could and should be reserved for those messages that require it. Two bits from the CAN identifier field could be reserved, without significant consequence for arbitration, for toggling the encryption and/or authentication on or off for a message.

The use of a timestamp for the authentication aspect of the cryptographic design is proven to be problematical despite the use of the NTP time synchronization protocol to try to address this issue. The use of a counter for the encryption scheme also has limitations, as the way that it is currently used assumes that it is always increasing; however, a node that does not receive a message with an updated counter value may issue a message of their own with a lower, previously used value. Having both a counter and a macTag in the CAN frame has serious negative consequences for the size of the counter, the macTag, and the data frame; employing just a timestamp instead of a counter would avoid this problem but would introduce those issues revealed when the timestamp is used for authentication. Further investigation is required to determine a reliable method of monotonically and consistently updating a counter/timer over the CAN network.

The effect of the OOB key distribution can be further investigated when the key server is running while the CAN bus is active. The key refresh can be triggered in advance based on a timer/counter (though this is then dependent on a reliable scheme for this) and the server-client relationship must be examined to allow the same keys to be distributed to different nodes in the same trusted group even if they have not requested the keys.
The recent announcement by NIST of a new SHA-3 family of hash functions offers the possibility of a hash function for HMAC that uses intermediate state sizes that could provide lightweight alternatives which would be interesting to explore in the constrained context of the CAN bus. This hash function has also been suggested for an authenticated encryption system that could replace the combined approaches used in this thesis. There are also lightweight stream ciphers such as WG-8 that could experimented with for the CAN bus.

The authentication and encryption schemes have been 'bolted' together for the overall design and use the arguably less secure MAC-then-encrypt approach. There are, however, dedicated modes for authenticated encryption such as Galois Counter Mode (GCM) and Counter Mode with CBC MAC (CCM), amongst others. A future research direction could be the investigation of these modes for CAN bus security, particularly for the CAN FD standard.

It is likely that the CAN protocol will persist as the network of choice in automobiles and some automated environment in the medium term, so continued efforts to investigate and improve the security of this network should be made.


MSCSED THESIS: CAN BUS SECURITY

Associated Press. (2013). Hackers find weaknesses in car computer systems. 


http://www.research.ibm.com/people/l/leiba/Papers/
Pervasive-Computing-in-Health-Care.pdf


things approach for managing smart services provided by wearable devices. 


MSCSED THESIS: CAN BUS SECURITY


MSCSED THESIS: CAN BUS SECURITY

http://www.isi.edu/~johnh/PAPERS/Farkas06a.html ((Special issue guest editors))


MSCSEDM THESES: CAN BUS SECURITY

area networks. In The 11th international conference on cryptology and network security, CANS.


Hoppe, T., Kiltz, S., & Dittmann, J. (2009). Automotive IT-security as a challenge: Basic attacks from the black box perspective on the example of
http://dx.doi.org/10.1007/978-3-642-04468-7_13


http://dx.doi.org/10.1109/MSP.2004.26

http://dl.acm.org/citation.cfm?id=1987310.1987333

http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html


Kargl, F., Papadimitratos, P., Buttyan, L., Muter, M., Schoch, E., Wiedersheim,


Mármol, F. G., & Pérez, G. M. (2012). TRIP, a trust and reputation


MSCSED THESIS: CAN BUS SECURITY

Retrieved 5th July 2014, from


