NTRCI

NATIONAL TRANSPORTATION RESEARCH CENTER, INCORPORATED
UNIVERSITY TRANPORTATION CENTER

U12: Data Security Solution for Trusted Truck® II

This project was funded by the NTRCI University Transportation Center under a grant from the U.S. Department of Transportation Research and Innovative Technology Administration (#DTRT06G-0043)

Dr. Itamar Arel, The University of Tennessee, Knoxville

Table of Contents

List of Figures

Executive Summary

Following the successful proof-of-concept demonstration of the Trusted Truck® project, an essential subsequent step, addressed by this project, pertained to the development of a robust data security infrastructure. The latter pertains to both the *confidentiality* of information exchanged, as well as *authentication* of the various parties communicating. Hence, the primary goal of this project was the development of a resource-efficient, tailor-fit data security protocol for the Trusted Truck® project. The solution has been coded and tested on real hardware and forms the basis for moving forward in providing data robustness and integrity for the Trusted Truck® network. Beyond the technical and scientific achievement, two students at the University of Tennessee (UT) were supported, thus enhancing the educational charter of both UT and the NTRCI University Transportation Center.

Chapter 1 – General Overview

Background

Following the successful proof-of-concept demonstration of the Trusted Truck® project, an essential subsequent step, addressed by this project, pertained to the development of a robust data security infrastructure. What makes the Trusted Truck® environment unique is the asymmetry that exists between the computational resources at the truck and the ones on the server side. We attempted to exploit this asymmetry so as to offer a cost-efficient cryptographic software solution that maps itself well to the underlying Trusted Truck® infrastructure. It is important to note that all data security solutions embedded within the developed protocol adhere to NIST-approved cryptographic standards.

Public key cryptography facilitates security mechanisms of crucial importance to the Trusted Truck® environment. It enables trucks and inspection stations to establish secure communications, and suggests authentication and signature methods that inherently cannot be offered by symmetric key methods. To that end, data obtained from all nodes in the network, particularly the vehicles, is guaranteed to be accurate and fully secure by enforcing both strong authentication (i.e. user identification) and confidentiality (via encryption) on all messages exchanged.

For the system to be secure, the applications must be able to trust that the communication has been received unaltered and from a known/verifiable source. Thus, a fundamental requisite for achieving security is the ability to provide for data confidentiality and authentication. The advantages of public key cryptography (PKC) for data security are widely acknowledged and include resilience, scalability and decentralized management. To that end, the goal of this project was to leverage existing ITS security standards in the design, implementation and evaluation of a comprehensive data security solution for the Trusted Truck® project.

Project Team

Dr. Itamar Arel – Principal Investigator (PI) for the project. Primary responsibilities included definition of the research direction and goals as well general schedule and management oversight of the research team. Moreover, all correspondence with Volvo, particularly with regards to exchange of technical information, was the responsibility of the PI.

Srinivasa Anuradha Bulusu (Anu) – was a graduate research assistant, primarily responsible for the cryptographic protocol development and core implementation. She is a Masters student of computer engineering in the Department of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville. She received the bachelor degree in electronics and communication engineering from JNT University, India.

Scott Livingston – was an undergraduate research assistant who has helped in the code optimization and demonstration platform development, supporting Anu in her research work pertaining to this project.

Dr. Riheng Wu – was a postdoctoral research associate working on this project for a several months during the commencement of the effort. His work focused on researching various cryptographic schemes considered for the data security protocol developed.

Project Description

The project involved the design, implementation and testing of the data security solution discussed above. The following are threat models and respective security solutions which were of primary focus:

(1) Explicit truck and driver identification; the first and foremost goal of this effort was to provide robust identification by the inspector of both the truck and its driver. This was achieved by requiring the truck to provide information which is digitally signed using unique private key(s).

(2) Prevention of tampering or replacing of sensors; truck sensors, such as those pertaining to the brakes system, authenticate any measurement data transmitted (via the truck's computer) to the inspection station.

(3) Prevention of impersonation attacks; replay attacks are inherently avoided.

(4) Cargo Identification; efforts were made to provide for explicit cargo identification valid through the entire route of the truck.

The data security protocols have been designed so as to be embedded in software located on the truck (On-board Unit – OBU) as well as at the inspection site (Roadside Unit – RSU). From a communications perspective, the solutions are seamlessly integrated within messages exchanged over the wireless network, such that no additional dedicated hardware is required. Moreover, effort has been made to reduce the computational load on the OBU to facilitate cost-efficient solutions.

Chapter 2 – Primary Research Goals and Schedule

This project aimed to provide the Trusted Truck® effort with a much-needed data security infrastructure. Augmenting cryptographic standards, the solutions developed have substantial impact on guaranteeing the continuing success of the project. Potential users of the outcome of this project include both public and private parties involved in achieving the Trusted Truck® vision. Without secure credentials, the Trusted Truck® concept is infeasible.

During the first three months of the project, the work has focused on developing a formal set of security requirements which addressed the primary needs as well as threat models that characterize the Trusted Truck® environment. During the next two months, a detailed study of the DSRC technical recommendation on security infrastructure was conducted; from it was ascertained that a solid standard has not been established. During the subsequent two months, novel security services that address the unique attributes of the Trusted Truck® environment were devised, in as much coherence as possible with the DSRC framework.

As a result of correspondence with Volvo staff members, it was decided that the appropriate manner by which the security protocol is to be implemented is within the context of TCP/IP. By coding the protocol in standard ANSI-C and using the TCP/IP protocol stack, we were able to offer a generic, platform-independent solution that can be utilized in a variety of OBU platforms. Moreover, such an approach allowed Volvo to view the data security services as a layer within their software solution.

The successful completion of this one-year effort is expected to provide a solid platform on which future Trusted Truck® developments can be made. Field trials will help establish a practical software package that can be easily incorporated within existing equipment. In an indirect way, data security will assist in safety provisioning, as it will make it much harder for malicious parties to intervene with legitimate ongoing communications.

Chapter 3 – Project Deliverables

The key challenge in this project was to develop a security protocol that would be asymmetric in its nature in the sense that the computational burden would shift from the OBU to the RSU. That has to be done while adhering to the public-key cryptographic needs of the protocol, particularly with regards to digital signatures and authentication. The result was a discovery at the mathematical level, which translated into an implementation of the cryptographic functions in a way that significantly alleviated the computational load from the OBU.

Security Protocol Basics

Figure 1 depicts the general information flow in the security protocol developed. From the perspective of the communication protocol the primary parties involved are the truck, the stations and the Certificate Authority (CA). All communications between the truck and stations (weigh station, departing station or arriving station) are achieved using Public Key Cryptography and digital signatures. The following provides a brief overview of the process involving the truck and stations, as well as discusses the role of CA.

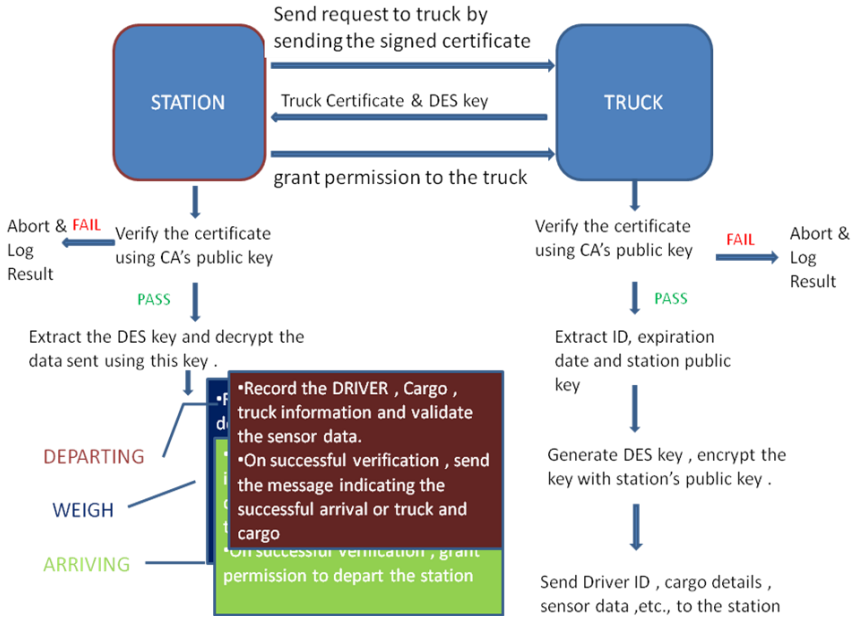

Figure 1. Illustration. Information Flow in the Trusted Truck® Data Security Protocol.

The truck is assumed to host the OBU while the departing/weigh/arriving station(s) are assumed to host resourceful workstations running the RSU software. For simplicity, we shall use the terms truck and station interchangeably with OBU and RSU. In Public Key Cryptography, the users (here the truck, station and CA) have two different keys – a public (commonly known) key used by the public, and a private key known and used only by the user. While encrypting the data, the user makes use of the public key while in the decryption process a user employs the private key.

As such, encrypting messages can be done by anyone (using the public key) while decrypting them requires a (secret) private key.

To generate a digital signature, a reverse order of key usage emerges since the user employs the private key as the means for signing a message while the public key is used to verify the authenticity of the signature. The public and private keys of the truck, station and CA are generated and approved by the CA. To that end, the CA must be an entity that is objectively referred to as credible by all parties wishing to communicate. The CA public and private keys are 1,152 bits long (144 Bytes), while the truck and stations use keys of size 1,024 bits (128 Bytes). The first software package is the one running on the CA and is responsible for the generation of all keys. Following the generation of keys, the CA distributes its public key to all trucks and stations, for future reference, as will be explained in the following section. The CA also signs the truck/station IDs, expiration date and their public keys.

Each truck is loaded with the CA's public key, in addition to the truck's public and private keys. The basic process involves establishing a session key between the truck and the station that is only known to these two participants. This session key is used to encrypt (using standard DES) the messages exchanged. The successful establishment of the session key will guarantee to all parties that all are authenticated, i.e. their identities are correct. A new session key is generated at the beginning of each communication session. All transactions with the station are recorded in the log file, while the truck software package initiates message exchanges.

All stations (departing, weigh & arriving) should be preloaded with their respective public key and private keys, as well as the CA public key. The particular nature of the message exchanged with the truck depends on the type of station it is communicating with. Transactions with each truck are recorded in a dedicated log file.

Technical Specifications of the Protocol

Generally speaking, the nature of the interaction between the truck and each of the stations is identical. What differs is the timing during which the specific message correspondence takes place. It is assumed that for all communications between the truck and a station, the session is initiated by the truck. Based on these assumptions, we summarize the protocol flow as follows:

1. Truck and station software packages are invoked
2. Log files with the ID and current timestamp are generated at the truck as well as at the station.
3. The truck is assumed to initiate the communication process or the message transfer process.
4. The station sends its certificate to the truck, the truck verifies the received certificate using (preloaded) CA's public key.
5. Should the verification fail, the truck reports the error and aborts, otherwise it extracts the ID, expiration date and the station's public key.
6. Send the encrypted truck's session key and the truck's certificate to the station.
7. Then send the required encrypted messages including driver ID, cargo details, truck physical condition and other related information.
8. The station receives the encrypted truck's session key and verifies the truck's certificate.

9. Should the truck's certificate be found invalid, the station reports the error and aborts; otherwise it extracts the vehicle ID, expiration date and truck's public key and decrypts the received message.
10. Once all details are verified, the station sends a validation message
11. The session with the truck is terminated and the results are logged in respective log files.

Scientific Contributions

As stated above, the truck's processor (e.g. ARM based) is assumed to have limited computing capabilities. In particular, the processor is not expected to generate an RSA signature in real time. This introduces a fundamental difficulty in authenticating the truck by any ground station, since public key-based authentication inherently has a signature generation ingredient, needed to be performed by the truck. We resolve this issue by embedding the truck's authentication within a mutually authenticated key-transport procedure, performed between the truck and ground stations. Here, the truck's computational efforts related to the authentication are performed off line, prior to the actual communication session conducted with the ground station. (The truck can perform its calculations on the way to a ground station, and at any time prior to departing from the base station.) To prevent re-play attacks, the truck has to perform this operation (off line) for every communication session.

This approach has implications on the relative sizes of the fixed keys (public and private) used by the various parties. These considerations are applied by the CA, who issues these keys. To further ease the computational efforts at the truck, an innovative implementation of the Chinese Remainder Theorem was devised, utilizing the fact that the transported key is 64-bits (DES key) processed by 512-bits public key cryptographic keys.

In this project all operations on the truck assume to employ a 16-bit processor and all stations, including the CA, host 32-bit processors. The private and public keys of the truck as well as the station are 128 bytes in length. The different key constraints and the computation effort on the truck and station are described as follows:

- The values of the keys generated for the truck are always less than the keys generated for the station. This is handled by the key generation package at the CA – which is incorporated so as to reduce the computational load on the truck in comparison to the the station.
- All modular exponentiations on the truck use the Chinese Reminder Theorem (CRT), whereas the calculations at the station do not. Exploiting the Chinese Reminder theorem reduces the computation effort by a factor of 4. This reduction in computation has been derived and verified for all modular exponentiation operations in this project.
- All trucks perform session key generation required for the offline authentication (16 bit processing is assumed). This offline-key generation is done for every communication session prior to approaching a station, thus saving substantial time. When the truck approaches the station, it verifies the station's certificate and simply transmits the encrypted signed key to the station. The major mathematical operations performed by the truck are
  - Offline Session key generation – $L = V^d \bmod n$ (using CRT) consuming approximately 0.12 to 0.14 sec
  - Certificate Verification - $R = S^3 \bmod n$ consumes at most 0.12 to 0.14 sec
  - Online-key generation - $C = L^3 \bmod n$ consumes at most 0.12 to 0.14) sec

- On the other hand, the stations perform all operations online. The major mathematical operations performed at the stations are:
  - Certificate verification – $R = S^3 \bmod n$
  - Session key recovery and extraction = $V = [C^d \bmod n]^3 \bmod n$ (without requiring the use of CRT)
  - Modular exponentiations (without CRT) consume approximately 0.2186 sec
- In the above calculations, the values utilized at the truck are shorter than those at the station, derived from the fact that the truck's public and private keys are shorter than the station's public and private keys.
- Mathematically, as the operation $B = A^3 \bmod n$ involves only one modular computation, it takes much less time when compared to the operation $B = A^d \bmod n$. Hence, when the truck commences communication with the station, it is not required to spend much time on calculation.

Keeping in mind that the trucks use a 16-bit processor, the computational load is greatly reduced at the truck by incorporating the CRT theorem and the various key constraints. As the time consumed by the offline session key generation is much less than that taken by a truck to travel between the various stations, the key generation process is substantially shortened.

Chapter 4 – Volvo Partnership and its Future Role

As indicated above, this project is a collaborative effort between researchers at the University of Tennessee and technical staff members at Volvo. The PI has been in correspondence with Volvo staff members and has received positive initial feedback on security protocol developed as well as the general need for a robust data security solution within the Trusted Truck® framework. Currently, the PI is in the process of passing the code developed to Volvo so that they can begin integration of the security protocol within the Trusted Truck® software packages.

The primary next three steps, with regards to the partnership with Volvo, include (i) integration of the security protocol (Volvo lead); (ii) development of the comprehensive framework that involves message servers, the TTMC and government back office communications infrastructure to support the overarching vision of the Trusted Truck® project; (iii) development of a physical system demonstration that will exhibit the overall system, initially in a laboratory setting, followed by an in-field demonstration.

Appendix A: Trusted Truck® Cryptographic Service Protocol

Introduction

This appendix provides a detailed description of the cryptographic procedures taking place at each of the participants in the Trusted Truck® network. There are five participant types: the certificate authorities (CA), truck, departing station, weigh station, and arriving station. It is assumed that the CA is unique. There may be arbitrarily many trucks and stations. The **certificate** of any participant, except the CA, is defined to be the expiration date (8 bytes; MSB: = 0), ID (8 bytes; MSB: = 0), public key n(128 bytes), and the CA signature of the preceding three values; thus, certificate length is 288 bytes. Only the truck uses a 16-bit architecture. All certificates, p, q, and the respective keys are generated on a 32-bit architecture.

Certificate Authority

The certificate authority (CA) generates RSA keys for all participants. In addition, it signs each participant's ID, expiration date, and public key. The CA public key is downloaded by the security officer to each station and truck when the private parameters are downloaded. The CA private and public keys (denoted d_{CA} and n_{CA} respectively) are generated by this package. Each is 144 bytes, facilitating a signature on 8+8+128 bytes. For a given participant (truck or ground station) ID and expiration date, the CA will:

1. (a) In case participant is a truck, generate p, q, u (for Chinese Remainder Theorem acceleration; 64 bytes each), *dp*, *dq* (private key; 128 bytes), and n_T (public key; 128 bytes); or
   (b) In case participant is a station, generate p, q (64 bytes each),d_S (private key; 128 bytes), and n_S (public key; 128 bytes),
2. Sign expiration date (8 bytes; MSB := 0), ID (8 bytes; MSB := 0), and *n*,
3. Combine results (including expiration date, ID, CA signature, n_{CA} CA public key) into a single deliverable.

The CA software should simply take as input an ID string and expiration date and output a single file to be copied on-site to the participant hardware. This new file will henceforth be referred to as the "new participant package."

Truck Operations

For each truck, a new participant package should be installed at a location to be determined, dependent in particular on the operating system in use on the truck hardware.

A new DES key K, called the "session key," will be generated and used for each transaction. To avoid delays, a single DES key K (8 bytes) should be generated, signed with the truck private key d_T (using RSA; result is 128 bytes with MSB := 0; call it D), and stored (for later use) upon loading the new participant package from the CA and immediately following completion of any transaction.

All communications with stations will be recorded in a log file; this log file could either be determined automatically based on local settings (e.g. truck ID and expiration date) or selected by some configuration file. In a typical transaction, the truck will:

1. Receive station certificate,
2. Use (preloaded) CA public key n_{CA} to verify station certificate,
3. If verification fails, log result and abort; otherwise, continue,
4. Extract ID, expiration date, and station public key n_s,
5. If certificate has expired, log result and abort; otherwise, log station ID and expiration date and continue,
6. Encrypt the signed (by the truck) session key D with the station's RSA public key,
7. Send truck certificate (288 bytes) and (signed and encrypted) session key (128 bytes),
8. Begin safe transactions (all sent items are encrypted with K):
   (a) send driver ID (8 bytes),
   (b) send conditions of sensors (8 bytes),
   (c) send cargo details (8 bytes),
   (d) receive station response,
9. Terminate session and log result.

As stated above, the truck should now immediately create a new, signed DES session key D in preparation for the next transaction.

Ground Station Operations

For each ground station, a new participant package should be installed at a location to be determined, dependent in particular on the operating system in use on the station hardware. As in the truck specification, all communications with trucks will be recorded in a log file; this log file could either be determined automatically based on local settings (e.g., station ID and expiration date) or selected by some configuration file.

In a typical transaction, a station will:

1. Send station certificate (288 bytes),
2. Receive truck certificate and (signed and encrypted) session key(128 bytes),
3. Use (preloaded) CA public key n_{CA} to verify truck certificate,
4. If verification fails, log result and abort; otherwise, continue,
5. Extract ID, expiration date, and truck public key n_T,
6. If certificate has expired, log result and abort; otherwise, log truck ID and expiration date and continue,
7. Recover mutually authenticated DES session key K,
8. Begin safe transactions:
   (a) Using K, decrypt received truck driver ID (8 bytes), conditions of sensors (8 bytes), and cargo details (8 bytes),
   (b) if departing station,
      i. validate driver; on success, send “driver is entitled to drive this truck” message (8 bytes),

ii. validate cargo; on success, send “truck is entitled to carry this cargo” message (8 bytes),
iii. validate papers; on success, send “papers checked and valid” message (8 bytes),
iv. validate sensor readings; on success, send “sensor readings are valid” message (8 bytes),
v. send destination (8 bytes),

(c) if weigh station,
i. record truck details; on success, send message indicating “details were recorded” (8 bytes),
ii. validate sensor readings; on success, send “sensor readings are valid” message (8 bytes),
iii. measure and record weight; on success, send ``measured weight'' message (8 bytes),

(d) if arriving station,
i. record truck details; on success, send message indicating “details were recorded” (8 bytes),
ii. validate sensor readings; on success, send “sensor readings are valid” message (8 bytes),
iii. log successful arrival of truck and cargo,
iv. send “truck and cargo arrived safely” message (8 bytes),

9. Terminate session and log result.