2010 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
MODELING & SIMULATION, TESTING AND VALIDATION (MSTV) MINI-SYMPOSIUM
AUGUST 17-19 DEARBORN, MICHIGAN

Importance of Secure Real Time Information to an Operator Using Robotics

Ramie Phillips III
Pure Entropy Technologies, llc.
Orion, MI

ABSTRACT

Information is critical to successful deployment and operation of unmanned vehicles. The increasing use of unmanned vehicles in modern conflicts has substantially increased the strategic and tactical value of these vehicles and the information they gather. It is now common public knowledge that the video streams of some predator drones were unencrypted and militants were able to use cheap commercially available software to intercept these feeds. This is an example of security as an afterthought. Encryption and security are critical to unmanned systems and should be implemented early in the development process. This paper explores some of the issues related to encryption and security of unmanned vehicles and communication.

INTRODUCTION

From the Wall Street Journal: "Militants in Iraq have used $26 off-the-shelf software to intercept live video feeds from U.S. Predator drones, potentially providing them with information they need to evade or monitor U.S. military operations. Senior defense and intelligence officials said Iranian-backed insurgents intercepted the video feeds by taking advantage of an unprotected communications link in some of the remotely flown planes' systems."[1]

This is an example of the absence of encryption between unmanned vehicles and operators. If a militant can use a simple $26 commercial piece of software to receive drone video feeds, imagine what an opponent with a large technology budget and groups of engineers is capable of compromising. Encryption is the solution to this problem. This seems to be a simple solution. However, actual implementation is difficult and there is much to consider when designing and installing a complete encryption solution. Encryption is only as strong as the manner in which it is implemented.

THE PROBLEM

Encryption for data in motion has many complexities. One of the first requirements is that the data remain encrypted from the generator of the data to the receiver of the data. This is known as end-to-end encryption and offers the greatest level of security. It must also be assured that the data is originating from the expected sender and that the data has not been modified in transit.

Unmanned vehicles can communicate many different forms of data, such as localization, health, video and sensors. This type of data is typically from vehicle to operator. However, ideas like collaborative sensing and distributed mapping may require transmissions from vehicle to vehicle and this should be included in the design if needed. These types of sensor data transmissions are typically unidirectional. Control of the vehicle through vector or waypoint commands highlight the need for bidirectional communication and low latency. A well designed encryption system will handle bidirectional communication as well as being data agnostic.

The encryption system should be designed to have a minimal effect on vehicle system resources, including power, processing, and communications. Another common requirement is for the system to be transmission media agnostic. For example, the actual radio, light, sound or other method of data transmission will not affect a well designed encryption system.

An encryption system needs to have secure methods of managing and transmitting keys with minimal or no intervention from an end user. It must also possess secure

algorithms that do not place undue loads on vehicle resources.

In summary, an encryption system should be media agnostic, data agnostic, fast and secure. One way of ensuring this is to include an encryption system in early planning phases of unmanned vehicle systems.

DESIGN CONSIDERATIONS

We must first determine where in the OSI model is the best place for encryption. It turns out that it is dependent upon requirements of the system.

Application	Application integrated encryption
Presentation	
Session	
Transport	
Network	Routable encryption
Data Link	Non-routable encryption
Physical	No encryption, media agnostic requirement

Table 1: OSI model guidance.

Different operating systems have different interfaces to encryption which is important when designing an encryption system that operates at the network or data link layer. Linux uses a TAP driver to access the data link layer and a TUN driver to access the network layer. Windows uses the NDIS interface to accomplish the same access. It should be noted that Windows XP and prior uses NDIS5, and later Windows operating systems use NDIS6. The general idea with these systems is to create a virtual device and use this device to process the frames and packets as they move through the OSI model.

When integrating encryption into the application layer it is most commonly distributed as a library that can be compiled into the application. Another option may be as a standalone executable or in a Windows environment as a Dynamic Link Library (DLL).

When making the choice between Network/Data Link encryption solutions and Application level encryption there are some basic considerations to make. The following table poses some of the questions to consider.

Question	***Yes***	***No***
Does all data in and out of the system need security?	Network/ Data Link	Application
Is the system resource constrained?	Application	Network/ Data Link
Is the data being received by multiple recipients?	Network/ Data Link	Application

Table 2: Encryption placement guide.

After the decision has been made on where to place the encryption the type of key management and encryption algorithms are the next logical choices.

Key management generally falls into 2 categories, those with a trusted third party and public key infrastructure (PKI) and those without. The primary advantage to using a PKI is the ease of creating signed messages. This means that the originator of the message can be discerned securely. However the inclusion of a PKI adds complexity to the encryption system and a trusted third party must be available, which often adds great complexity and cost.

In systems without PKI the signing of messages is much more difficult. However, the lack of needing a trusted third party makes the key management much easier to use in actual implementation.

Finally a symmetric encryption algorithm must be chosen to do the actual data encryption. These fall into 2 basic categories, stream and block. Streaming algorithms tend to be faster and allow for creation of a buffer in resource constrained environments. Block algorithms process the data as part of the algorithm and in secure implementations require the results of the prior encryption. This property precludes the use of a buffer to utilize resources and other time than on demand.

OUR SOLUTION

Pure Entropy has addressed these issues through multiple implementations, and each one is slightly different. Our specialty is designing functional encryption systems that meet the specific needs of a project.

Since the vast majority of the encryption within a system is done with symmetric algorithms, Pure Entropy designed a symmetric algorithm called JUMBLE. JUMBLE can be run as either a block or stream encryption system. It possesses either a 1024 or 2048 bit key space, which far exceeds the industry standard AES 128 or 256 bit key. JUMBLE also is

designed using simple mathematics that minimizes system resource consumption. JUMBLE takes less computing cycles per byte of output than AES. This allows the battery and processor to be more available to do other critical mission tasks.

JUMBLE has been implemented at the application layer. It is available as a Linux library, Windows library, or a Windows DLL. It also can be implemented using a command line interface. However as a standalone application it is primarily used when encrypting data at rest, not data in motion.

JUMBLE has also been implemented at the Network/Data Link layer in Windows XP embedded. This particular implementation is in the form of a NDIS driver and performs complete end to end key management without a PKI. This allows for secure ad-hoc connections between devices.

REFERENCES

[1] S. Gorman, Y. Dreazen, and A. Cole, "Insurgents Hack U.S. Drones", Wall Street Journal, 2009.