Sam Utley
29 Oct. 2012

ALERT2 Encryption for Self Reporting Protocol: A Proposal

I believe the following proposal would allow encrypted and unencrypted self report messages to coexist on the same network. By not encrypting the control byte and time stamp, no knowledge of encryption will be required on any other layer.

The ALERT2 Application Layer Protocol Specification has declared the Self Reporting PDU as a concatenation of one or more data reports prepended by a control byte and a UNIT2 timestamp. Each data report uses a Type, Length, Value (TLV) structure, where the Value field may recursively embed lower level TLVs.

[Control] [Time Stamp] [Type] [Length] [Value] [Type] [Length] [Value]

This document proposes the creation of a fifth Report Type to identify a Self Report sub protocol for the delivery of encrypted message blocks. Herein, this new Report Type will be referred to as “Type 255” or “Encrypted Message Block”. It has a recursive TLV structure with the upper level Report Type (T) and Report Length (L) syntax remaining the same as described in the current Application Layer Protocol Specification. The Report Value (V) contains a second/sub TLV that specifies the encryption type, length of message prior to encryption, and encrypted message. At this point, the encrypted message would contain a concatenation of one or more General, Tipping Bucket, or Multi-Sensor reports.

If accepted, there would be five Self Report Types defined.

- Type 1: General Sensor Report
- Type 2: Tipping Bucket Rain Gage Report
- Type 3: Multi-Sensor Report – English Units
- Type 4: Multi-Sensor Report – Metric Units
- Type 255: Encrypted Message Block

The Report Value (V) for the Encrypted Message Block report type contains a recursive TLV structure. The first byte serves as the type (T), indicating the encryption algorithm used. The second and third bytes serve as length (L), specifying the length of the message prior to encryption. This length is needed to deal with any padding bytes added by a block encryption algorithm. Though length could theoretically be [0xFFFF], in practice one should be careful not to create an overflow condition. For example, if using the Blowfish Algorithm (see below) the length of the un/decrypted content should never exceed [0xFFF8].

An Encrypted Message Block would be described by the following recursive TLV structure. As shown, the recursive structure utilizes 1 byte to specify the type of encryption, 2 bytes to for the length of the un/decrypted message, and up to 65528 bytes for the encrypted message. For illustration, a General Sensor Report, Tipping Bucket Report and Multi-Sensor Report has been

shown as the content of the encrypted message.

T (1 byte)	L (1-2 byte)	"Encrypted Message Block" Report (1-65535 byte)									
		Encryp Type T (1 byte)	Decryp Len L (2 byte)	Encrypted Message V (1-65528 byte)							
				T	L	GSR	T	L	TBR	T	L	MSR

An alternative, but in my opinion less attractive, proposal is that Report Value (V) starts with a two byte header. The two most significant bits of the header indicate the encryption protocol code used. The next fourteen bits note the length of the content portion of the encrypted message. Again, this length is needed to deal with any padding bytes added by a block encryption algorithm.

It is proposed that the Blowfish Encryption Algorithm serve as the first cipher supported. It is a relatively strong encryption algorithm (assuming strong keys are used) with relatively low processing overhead. The algorithm is public domain and free for use in any application.

When using Blowfish, the encryption type would be set to 1.

Blowfish is a symmetric key algorithm, meaning that the same key is used to encrypt and decrypt the message. This key can be 8 to 56 bytes long and would be pre-shared between the points of encryption and decryption. In practice, it would be easiest to use the same key throughout the network. Each device that wishes to decrypt a message will need to know the key. However, it would be possible to use more than one key in a network. It would also be possible use unique keys for each station. But again remember, a decryption device (e.g. base station software) will need a more sophisticated key management system for such a setup.

Blowfish is a block algorithm meaning that the message to be encrypted must be evenly divisible by a predetermined block size. The original message will be padded with nulls (generally) to round it up to the next evenly divisible block size. Blowfish uses an eight byte block size. The minimum message size for encryption is 8 bytes; all messages encrypted must be evenly divisible by 8 bytes. For example, if the message to be encrypted was a General Sensor Report for battery voltage (S=8, FL=UINT1, V=125) it would be padded with three null bytes during encryption [0x010308117D000000].

Following is an example. The unencrypted message shown has been selected from Appendix 1, Figure 4-1, of the ALERT2 Application Layer Protocol Specification version 1.2. The encrypted message contains the same data, wrapped in the Encrypted Message Block report type, using th Blowfish cipher and a 8 byte key of “password”.

Field	C	T	L	S	FL	V	S	FL	V
Unencrypted, Hex	70	01	0A	12	34	41 00 A3 D7	13	22	02 76

Field	C	T	L	T	L	V
Encrypted, Hex	70	255	13	01	00 0C	65 A7 AB 72 0D 95 B3 6F CB 48 D5 54 76 7B D8 A4