3M United Kingdom PLC
3M Centre
Cain RD
Bracknell
Berkshire
RG12 8HT
United Kingdom

Vehicle Enforcement System (VES) Software Interface Specification

VES SOFTWARE INTERFACE SPECIFICATION ISSUE 1V

The information contained in this document is confidential and is the property of 3M United Kingdom PLC. The contents must not be disclosed to any third party without the express written permission of 3M United Kingdom PLC.

AMENDMENT RECORD

ISSUE	DATE	BRIEF DETAILS OF CHANGE
1a	07-Jun-05	Initial version of VES specification – derived from TFL system.
1c	24-Aug-05	Amended to reflect continuing development
1d		changes to data structures, detail on shared key
1e	01/10/05	Changes for HGV enforcement system. Initial notes on data entry
1f		Section on passwords, nvt binary interface
1g	30/10/05	reorganisation, further detail, addition of references and glossary
1h		Amend pull protocol to enable explicit delete
1i	09/12/05	Data structure changes.
1j	11/12/05	STC comments incorporated
1k	19/01/06 24/02/06	updates to diagnostics, differences between basic and full configuration. Updates to secret entry
1l	19May06 jul 06 aug 06	Description of dummy record generation and trace facility Correction to heartbeat additions to exception list
1m	25aug06	integrate changes, convert to odt and release as draft
1n	15Sep06	Add context order options Merge in radar control addendum
1o	20Oct06	Add sections on SPIKEstore interface - initially as a pointer to an external addendum. Latched exception across a session.
1p	Mar 08 May 08 June 08	GPS time synchronisation Batched Summary records trajectory support summary record throttle exceptions updated ER unique encryption documented
1q		Internal extended encryption and control for enforcement applications
1r	Feb 09	GPS position information
1s	15/07/09	Add secondary time difference to batched summary records
1t	01/08/09	Add encryption to message channel & reformat
1U	05/08/13	Update logos and titles in documentation for 3M
1v	20/12/13	Amend to be generic to 372 and 382 camera system

The overall issue status of this document is the latest issue shown in the table above.

Author: Graham Wood

Table of Contents

1 Document Overview

This document describes the full VES interface specification. Some areas of this specification are not yet implemented. Other areas may not be relevant to specific OEM applications.
The VES system is offered in two generic variants. Basic VES offers a minimal system with no session control, authentication or encryption. Full VES offers session control, authentication and encryption. Either version may be offered with or without software to buffer records on compact flash.

This document describes a wide selection of interfaces and protocols which make up the VES suite. It is unlikely that any one application will require or make use of all the facilities described. The VES system may be built in specialised form for specific OEM requirements.

Consult in detail with 3M and the authors to ensure that the implementation and configuration in use is most appropriate to the specific application.

Where possible, this document will indicate which facilities are not present in basic VES. Additional notes will be available for specific OEM builds.

1.1 Introduction

This specification describes the software interface available within the 3M System ANPR unit to provide generic tolling and violation enforcement facilities. This system is a 3M ANPR camera with software dedicated to the task of providing image and vehicle tag data for congestion charge, road tolling and other violation enforcement applications.

This document should be read in conjunction with your respective ANPR camera manual.

The interface between the Instation and the ANPR Outstation is defined to provide the following functions:

- ANPR Outstation time synchronisation with the Instation
- ANPR Outstation time synchronisation monitoring
- ANPR Outstation configuration data download from the Instation
- ANPR Outstation software download from the Instation
- ANPR Outstation reporting vehicle summary data individually or by batch
- ANPR Outstation reporting vehicle image data
- ANPR Outstation reporting status data on a regular basis
- ANPR Outstation reporting status alarms
- ANPR Outstation switch over between primary and secondary instation

The violation (evidential) record connection and data stream may be authenticated and encrypted. Steps may be taken to protect the integrity of the camera system whilst violation records are being collected and transferred.

1.1.1 Physical configuration

The camera will need a trigger in order to commence a plate capture sequence. This trigger is generated internally when the hardware plate finder detects the presence of a potential plate in the field of view. Alternatively when interfaced with a DSRC systems or other road monitoring equipment an external trigger may be supplied. This external trigger may be a hardware open/close contact or may be a message sent to the camera over ethernet or a serial link.

On receipt of trigger event the camera will capture a small (configurable) number of image fields at a selection of exposure values as the vehicle passes through the field of view of the camera. In addition, at approximately the same time, one image will be captured from the overview camera (if present and enabled). From the license plate image fields the camera will select the most suitable for further processing. The selected field will be processed to locate and read the best candidate for a license plate. The selected field will have an associated time stamp. If no plate is found the camera may still forward the overview image. Where no separate overview camera is installed, the IR camera may have one exposure in its cycle configured to generate an overview field. Optionally further context images may be added to the event data set.

These context images may be captured before and after the main event.

If an enforcement session is active the camera forwards a secure evidential record containing all data and images to the host server.

A short summary record may be forwarded to the host server on all events.

The camera ALPR unit is equipped with a 100baseT Ethernet connection and supports TCP/IP V4 protocols. The camera will be connected to the VES server and management systems via this Ethernet connection.

The VES server will provide a dedicated application configured to accept communication and data transfers from the camera. The camera supports optional maintenance, control, message, diagnostic and exception channels.

The host system will require a mechanism to confirm that the camera ALPR engine is operating. To satisfy this need the camera may be configured to periodically forward a heart beat signal to the designated server. The heartbeat signal carries operational status information. If this function is enabled but the transmission is unsuccessful the camera can be configured to take recovery action.

The camera system may be configured to transfer all image events, or to transfer only those events requested by the host system. I.e. it can run in a data push or a data pull mode. When all events are to be transferred it is recommended that the camera pushes data to the server as this is the more efficient mechanism. When only selected events are required then the camera may be switched to use a pull protocol. The pull protocol may use ASCII or binary control messages.

2 General Configuration

This is described in broad detail in your respective camera user manual.

All 372 configuration options are set or modified either via a command line interface (CLI) or a web page interface. Commands to the CLI may originate either via serial or telnet connections or via script files. The system may be configured to run a script file automatically at start-up. This script file may be kept locally or may be copied automatically from the default FTP server

The full CLI interface may be access either via:

The debug serial port
A standard TELNET connection on port 23

The debug serial port will generate unsolicited messages reporting system behaviour. The exact messages generated will depend on various system configuration, logging and debug options. 3M therefore does not recommend that this port be used as a machine/machine interface, but that access be available for system configuration and for system software maintenance.

The 372 maintains a system event log reporting key system behaviour and may if required maintain a data log file (analyse.log) tracking image events. These files may be viewed via the CLI or transferred to a host system via ftp.

The following sections cover configuration options of particular relevance to the VES system.

For further details on the configuration and operation of your ANPR camera please refer to the respective camera user manual.

2.1 VES specific configuration

The specific processing route is selected by means of the CLI command:

```
>> set route ves
```

A set of dedicated CLI options exist.

command	parameter name	value	description	notes
`ves show`			show VES specific system parameters	
`ves set`				
	`host`		set host server ip address for host which will accept event records	1,10
	`alt_host`		Should the primary image host not respond to ping requests or fails to respond to open or transfer requests, then the system will assume that it is offline, and will switch transfers to this alternate host	10
	`camera`	`abcd`	name or identifier for camera. This is a short name, not a full site description. Length is limited to 21 characters. Format is TBD but must not include space or comma.	
	`lane`		name or identifier for lane. Length is limited to 11 characters. Format is TBD but must not include space or comma.	
	`store_host`		Set the IP address of the SPIKEstore module	14
	`store_account`		Set the account name on the SPIKEstore module	14
	`store_password`		Set the account password for the SPIKEstore module	14
	`store_path`		Set the target directory path on the SPIKEstore module	14
	`directory`	`/xxx`	Directory on local system used to maintain disk buffer of idiagnostic and exception files when host system is offline. Typically will be set to `dos:/ves` ie a subdirectory on the flash card.	12
	`cabinet`		ip address of cabinet controller. The camera will ping this address to ensure that the link to the cabinet is intact. Set this ip address to zero or empty to disable this function	11
	`router`		ip address of network router. The camera *may* ping this router (icmp ping) to establish that the local network is intact	11
	`security`		specify security level 0x00 – none (no authentication or encryption) 0x01 – generate HMAC of data with key 1 0x02 - generate HMAC, encrypt all data with key 2 0x04 - generate unique key block for each ER 0x08 - encrypt NVT channel 0x10 - encrypt message channel 0x80 - encrypt diagnostics during session	7,11
	`crypt`		(parameter removed) Encryption must be AES256	7,11
	`key_port`	`nnnn`	Port on which system will listen for session start request and on which key exchange will take place. Set to zero or empty to disable key exchange.	11

command	parameter name	value	description	notes
	image_port		set port on host system which will be listening for a connection for transfer of image records. Image transfer is disabled if this port is not set	1,10
	link_timeout	nnn	Close the image or summary connections if no transfers have occurred for this many seconds. Set to zero to disable.	2
	summary_port		set port on host system which will be listening for a connection for transfer of summary records. Summary record transfer is disabled if this port is not set.	10
	nvt port		command interface for machine-machine control between camera and lane controller or instation. The camera listens on this port.	10
	nvt_timeout	Nn	NVT interface timeout (seconds). If no requests or heartbeats are received for this many seconds then the socket will close and listen for a new connection.	
	nvt_mode		Select nvt operating mode: 0 – disabled 1 – ASCII interface 2 – binary interface	
	msg port		set the local tcp/ip port used by the message channel. The camera listens on this port.	3,10
	msg_timeout	nn	If no messages have been sent on the message channel for nn seconds then a keep-alive message will be forwarded to indicate to the host system that the message channel is still active.	
	hb_port		set the port number for the process on the host lpr server expecting heart beat messages. The host system listens on this port.	
	heart_beat	nn	period in seconds between heart beat messages. If set to 0, no heart beat messages will be sent.	
	not_alive	nn	if the heart beat process is running, (heart_beat > 0) then if the heart beat fails to send its message (i.e. unable to establish connection) for this many attempts, then the 372 will attempt recovery action.	
	trig_port		Port on which system will expect TCP/IP trigger messages. The camera listens on this port.	
	trig_timeout	nn	If a trigger port connection is open, then the camera will send a heartbeat out on the trigger port after this many seconds. (This is a simple heartbeat to indicate to the attached controller that the port is active.)	
	transfer	0xnn	set to select which images and messages are transferred to the server.	6
	debug	0x00	This is a bit field specifying the debug message which may be generated on the serial port or stderr port. The exact format is TBD	

command	parameter name	value	description	notes
	timeout		if no plate image is received on a camera for this number of seconds then a bit will be set in the status register. This may indicate a possible system failure e.g. camera fault or, may indicate that a lane is closed.	
	ping_interval		Interval at which background process monitors presence/absence of servers	
	ping_wait	nn	Number of milliseconds to wait for a ping response before deciding that the servers are off line.	
	nlog		Maximum number of images to buffer in flash disk, after which the oldest image will be overwritten	12
	file_size		Maximum size (in kbytes) of image dataset to buffer in flash disk. This parameter should be chosen carefully in conjunction with nlog to make best use of available storage. Any evidential records larger than this size will be discarded.	12
	ack_enable		Set bits to require that the host system acknowledge each transfer request.	8
	ack_timeout	nn	Number of ms to wait for an acknowledgement before automatically resending a record	
	ack_retry	nn	number of attempts the system will make to resend a record before giving up and discarding the data.	
	scan		Number of seconds between scans of flash disk buffer to check whether any files are waiting transfer to host system	
	threshold		Image events are discarded if plate read confidence is below this level	
	orientation	n	specify camera orientation as cardinal point where 0-N, 1-NW, 2-W etc	
	time_slot	nn	Diagnostics reporting is staggered from nominal period by this percentage	9
	context_count		number of supporting context images required (max 8 - default 0)	11
	context_offset		nominal time between context images (ms) See also capture parameter ctx_sample	11
	context_order	n	Set the ordering of context images. Images may be: 0 - split evenly before/after the trigger point 1 - all before the trigger point 2 - all after the trigger point For option zero there will always be an even number of additional context images. Options1,2 allow odd numbers of context images	11
	slot_age_limit	n	After a record has been in memory for this period (seconds) without successful transfer to the host system it will either be deleted, or transferred to the compact flash storage (if fitted)	

command	parameter name	value	description	notes
	`cf_age_limit`	`nn`	Set the maximum number of hours data may be retained in compact flash. Any data older than this will be deleted and not transferred to the host system.	
	`summary_rate`	`nn`	Set a control on transfer of summary records based on the outstanding count of evidential records. Set to zero to disable this control. Minimum active value:10 When set, if there are more outstanding ERs than this count, then transfer of summary records from backing strore will be suspended. A hysteresis of 10% is applied.	
	`radar_enable`	`n`	Set to 1 to enable the point speed radar interface.	`13`
	`radar_threshold`	`nn`	Set the point speed radar measurement threshold below which records will be discarded.	`13`
	`store_enable`	`b`	Set to 1 to enable transfer of data to the SPIKEstore module (if present and configured)	
	`max_batch_size`	`nn`	Batched Summary only. When the current batch reaches this size, the batch is completed and queued for transfer	
	`bsr_port`		The port to which batched summary records are directed	
	`bsr_nlog`		The number of summary record batches which are retained. When this number is reached, the oldest batch is overwritten	
	`bsr_interval`		The batch process interval (seconds) after which a new batch is started.	
`ves send`	`nnnnn fffff`		transfer evidential record with event_id `nnnn` and file_id `ffff` where `nnnn` is the hexadecimal sequence id generated for each event by the camera and reported in the summary record. `fffff` is the hexadecimal file_id reported in the summary record	
`ves del`	`nnnnn fffff`		delete evidential record with event_id `nnnn` and file_id `fffff` where `nnnn` is the hexadecimal sequence id generated for each event by the camera and reported in the summary record. `fffff` is the hexadecimal file_id reported in the summary record	
`ves diag`	`show`		show current diagnostics configuration – see specific section below	
	`set`		set specific diagnostic parameter – see specific section below	
`ves excep`	`show`		show current exception configuration – see specific section below	
	`set`		set specific exception parameter – see specific section below	

command	parameter name	value	description	notes
`ves stat`			Display statistics and histograms to indicate image latencies.	
`ves list`			Display list and status of image events in ram	
`ves loc`			set long form description of camera location	
`ves secret`			set shared secret (or passphrase) if required. Format is free text, maximum length is 64 characters. See below for details	11
`ves init`			Initialise or reinitialise file store system. Capture of image events is halted, all files in the file store are deleted, then a complete new set of files are allocated. This operation may take ten minutes to complete on a system with a 1gbyte device.	
`ves remove`			Remove all records from the flash file store	

notes

1) This designates the host server IP address which will accept the license plate data and images.
2) Once the camera has opened an image transfer connection it will hold the connection open for future transfers. If no transfers take place for this many seconds, then the connection will be closed
3) deleted
4) deleted
5) deleted
6) Other system configuration parameters must be set appropriately for each configuration. This parameter controls how files are transferred to the host system

`0x0001`	transfer patch	
`0x0002`	transfer full plate image	
`0x0004`	transfer front overview image	
`0x0008`	transfer context images	
`0x0010`	Require explicit delete	
`0x0020`	add text to patch & ir images	
`0x0040`	discard events below threshold	
`0x0080`	transfer events only on DSRC or instation request (ie select a *pull* protocol)	
`0x0100`	on tx failure buffer on disk	
`0x0200`	Enable transfer of M001 message	
`0x0400`	Enable transfer of M002 message	
`0x0800`	Enable transfer of warning message	
`0x1000`	Enable transfer of error messages	
`0x2000`	Transfer heartbeat as H001 message	
`0x4000`	Transfer session state messages	
`0x8000`	Enable transfer of summary records	
`0x10000`	Enable creation of short summary records	
`0x20000`	Enable transfer of batched summary records	

7) The image file transferred has an SHA1 HMAC authentication record appended. The authentication field or the whole data record (except for the first line of the header) may optionally be encrypted. Encryption key length may be selected. AES can be 128 or 256 bits. It is assumed that key_1 and key_2 will be the same length. Keys 1,2 are supplied as part of the session start protocol. OEM configuration may remove these options and force configuration to a specific mode.
8) Transfers on the image and summary record channels may be acknowledged. Sets bits according to which channels require the acknowledge:

0x01	require acknowledge on image record transfer
0x02	require acknowledge on summary record transfer
0x04	require acknowledge on diagnostic records
0x08	require acknowledge on exception records
0x10	Require ack on transfer of spikestore records
0x20	Require ACK on transfer of Batch summary records

9) With many camera outstations reporting regular diagnostics it is necessary to take steps to ensure that these regular communications from outstations are staggered in time. This parameter sets the percentage offset used by this site to communicate with the in-station. Thus for example if this station must send its diagnostic data every 5 minutes and this offset is set to 40 then this site will transmit data at every 5-min + 2-min
10)If any of these configuration options are changed the system may require a reboot before the change becomes effective. (Specifically, if a connection is open, then the new IP address cannot take effect till either the connection closes through normal reduced traffic flow or the connection is renewed as a result of a restart.)
11) This option will not be visible on basic VES systems
12)This option will not be available on system which do not have compact Flash support installed.
13) Options only valid and present on systems configured to support the radar interface. See the doppler radar addendum to this document.
14)Options only valid and present on systems configured to support the SPIKEstore distributed roadside storage module. See SPIKEstore specific manual and the addendum to this document.

2.2 Site identification

There are a number of separate site identifiers. There is a long site description indicating in user meaningful terms the location of the camera. This string is reported as part of the summary record and ER but otherwise is unused. It may be added to the images, but this is TBD.

Each camera has a separate camera identifier and lane identifier. These are short free format strings (which must not include space or comma) . They are generally “machine” identifiers.

The requirements call for IR camera and context camera identifiers. As the camera is a single item a single identifier is supplied to fill both needs.

Throughout this document there is a reference to site_id. This is formed from a concatenation of the camera identifier and lane identifier with an underscore between. e.g. camera C123 and lane 2 will have a site identifier C123_2.

1.1 Duplicate filtering

The system may be set to reject duplicate plates received on successive or near successive events. Duplicate filtering may turned off or turned on at selected levels.

`cap set duplicates`	n	where n 0 – duplicates rejection off 1 – duplicate rejection on. The comparison for “same plate” will consider very similar characters to be the same character (eg: B is considered equivalent to 8) 2 – duplicate rejection on. Matching uses common substring detection to assist in detecting duplicates where one image may have a portion missing (e.g. through obscuration by another vehicle or a pedestrian)
`cap set duplicate_level`	n	specify the percentage match between plates which will result in the plates being considered duplicates
`cap set duplicate_depth`	n	Number of entries in the duplicate history list
`cap set duplicate_age`	n	Maximum age (in seconds) of any entry in the duplicate history list. If a plate reappears before its entry expires, then its age is refreshed by the new occurrence.

The duplicate history has a historic depth of `duplicate_depth` plates. Plates in the history list will be discarded if they have not reappeared for `duplicate_age` seconds.

2.3 Time Synchronisation

This application requires that the license plates captured by a camera be matched with events detected by other systems. This is managed by time stamping captured images. For this to work the two systems must share a common clock.

The camera system will attempt to maintain local time and date via a connection to either a BSD / UNIX style SNTP (preferred) or a daytime server on a specified host machine at port 13. Most Unix and NT systems will provide this service or can load a service to do so. This service call may be disabled by setting the time service host IP address to 0. If this service is disabled, then the 372 internal clock / calendar will drift with respect to external time. Note also that event log message timestamps are taken from the calendar.

Local time may also be locked to a GPS time receiver providing NMEA records and a PPS (pulse per second) signal. When this is done the hardware tamper input is no longer available.

3M can if required recommend an SNTP server suitable for use with Windows NT or Win2k.

If a compliant SNTP server is used all times will be UTC. If a daytime server is used then all times will be set using the standard to which the daytime server is set.

Further details on time synchronisation may be found in the main configuration document.

NTP and SNTP protocols are documented in *RFC1305, RFC1769* and *RFC2030*.
The daytime protocol is described in *RFC0867*. In this application the daytime client will expect time in the form:

Day Mon dd hh:mm:ss Year

(Daytime protocol is not recommended, SNTP is preferred.)

Time server controls are within the system page of the CLI or web interface. Relevant controls are:

Command	**Parameter**	**Description**
`system set time_server`	`aa.bb.cc.dd`	Specify the server IP address to be used for the internet "daytime client" and SNTP requests. Set this to 0 to turn off calls to a time server
`system set alt_time_server`	`aa.bb.cc.dd`	Set the IP address of an alternate time server to contact if the primary does not respond.
`system set sntp_latency`	`nn`	set the worst case acceptable link latency to nn (decimal) ms. If after an sntp request is sent and a response received, the indicated latency exceeds this limit then the data is discarded and fresh attempt is made.
`system set sntp_window`		set the window within which time must be locked (ms). ie if set to 200 the system will attempt to main time lock to within 200ms of nominal.
`system set time_poll`	`nnn`	interval in seconds between time synchronisation events.
`system set sntp_max`	`nn`	If the internal time and the host time differ by more than `nn` hours (default 24) then a serious fault may exist which could be with the host time server. The system will not atttmpt a time correction. Set this parameter to 0 to always start the time correction process.

`System set sntp enable`	`0xnn`	`Set sntp and daytime enable. Default is 0x13.`
		`0x01 - enable sntp client`
		`0x02 - enable daytime client`
		`0x04 - enable GPS client`
		`0x10 - enable sntp server`

Low pass filtering is applied to time corrections. If the result a of a time correction change is still outside `sntp_window` then a further time correction sequence takes place. After 20 events the system will pause for `time_poll` before trying again. Initial time lock after system start when the internal clock may have drifted a few seconds may take several sntp attempts. If the time discrepancy is minutes or hours then several `time_poll` periods could elapse before full lock is established.

If `system set sntp_max 0` then the initial time correction is always applied without any low pass filter and whatever the correction required. This allows a system to rapidly achieve a synchronised status at system start.

Once the camera has achieved time lock, then an alarm indicating loss of time lock will not be raised until a period of twenty minutes has expired without a successful SNTP update.
A camera may also function as an SNTP time server. The camera will return "unsynchronised" to any clients until a successful lock is achieved.

Position information

When a camera is equipped with a GPS receiver position information may be recorded and transferred as part of the Summary and Evidential records. See GPS configuration details (documented elsewhere) for correct configuration of the GPS subsystem

2.4 Validation of system configuration

(See respective ANPR camera user manual for detail)

The camera will use active copies of configuration data stored in SRAM. However after an extended power fail (> seven days) this data may be suspect. Under such circumstance, the active copies will be refreshed from copies kept in the boot flash memory. A procedure exists to update boot flash copy. As part of the system validation process, the boot flash and the SRAM configuration details must be identical, and, in addition a reference CRC of this data and the program image files must also agree. The commands used to manage this system are:

Command	***Description***
keyfile update	copy key configuration files from sram to boot flash
keyfile restore	restore current set of key configuration files from boot flash copies to sram working copies
keyfile validate	Ensure that CRCs of key configuration files in boot flash and sram agree, and if they do agree, save a reference CRC in eeprom. This command requires a security code to complete.
keyfile show	Report eprom checksum and the application and configuration CRCs

There are two file sets comprising:

application	main application image gate array image font files anpr file	
configuration file set	configuration file access control list	

Procedures exist for automatic transfer and update of system files. These are described in an appendix to this document.

3 Sequence of events

This is a brief outline of system behaviour.

The camera system may transfer data either directly to the host system (a push protocol) or only on request from the host system (a pull protocol). When the majority of records must be transferred to the host system then the use of the push architecture is recommended as this will be the most efficient. A pull architecture helps to reduce the flow of data if the receiving end is carrying out de-duplication and only requesting specific records.

The VES system will normally be equipped with non volatile storage to provide local buffering, especially for a pull system where the pull request can be delayed. This will usually comprise a compact flash storage device of between 64mbyte – 2 Gbyte.

3.1 Push protocol

1) A trigger is received either from the external trigger input (serial or ethernet as for example from a DSRC system) or from the internal plate finder hardware. Ethernet triggers can only be recommended for systems with very slow moving traffic.
2) The camera start to look for a vehicle plate. The camera continues to look for a plate until the trigger "open" period expires or until the vehicle leaves the field of view. The open period defaults to about 150mS, but is fully configurable.
3) At the earliest opportunity within the exposure sequence the camera captures an overview image. (If this option is enabled). Further overview fields are captured as required during the vehicles transit through the field of view.
4) At the end of the open period or transit the ANPR engine processes the best license plate image found.
5) Summary and (if required) evidential records are built containing the specified images and other event data with appropriate authentication and encryption. This record is saved in an internal table with room for in excess of 100 events. If this table is full then the oldest record is overwritten or if buffering is enabled, it may be transferred to the non volatile store.
6) The complete event record is then transferred to the host system. If the host server is not contactable, the record is buffered in the non volatile store until it is either successfully transferred or overwritten when storage becomes full.
7) On completion of the event transfer an M004 message may be generated.

3.2 Pull protocol

1) A trigger is received either from the external trigger input (serial or ethernet as for example from a DSRC system) or from the internal plate finder hardware. Ethernet triggers can only be recommended for systems with very slow moving traffic.
2)The camera start to look for a vehicle plate. The camera continues to look for a plate until the trigger "open" period expires or until the vehicle leaves the field of view. The open period defaults to about 150mS, but is fully configurable.
3)At the earliest opportunity within the exposure sequence the camera captures an overview image. (If this option is enabled). Further overview fields are captured as required during the vehicles transit through the field of view.

4)At the end of the open period or transit the ANPR engine processes the best license plate image found.

5)Upon the reading of a licence plate, a Summary and an Evidential Record will be prepared with a common identifier.

6)The Summary Record will be placed in a queue to be sent to the Instation as soon as possible. Aged Summary Records will be stored in non volatile storage. The Evidential Record is placed in a queue and if not retrieved before it is aged, or deleted, it will be stored in non volatile storage.

7)The Instation will acknowledge receipt of the Summary Record. Upon positive acknowledgement it will be marked for deletion, and on a bad acknowledgement it will be resent.

8)The Instation will decide whether or not it wishes the Evidential Record to be retrieved. If it does so it will request transmission using the common identifier, else it will request deletion using the common identifier.

9)Optionally the Instation will acknowledge receipt of the Evidential Record. In the event of a failed acknowledge, the record will be resent.

10)Upon a successful acknowledge, the Evidential Record held in the memory queue will be marked as having been transferred.

11)Optionally the Instation may explicitly delete a record which has been successfully received.

12)As the number of Evidential Records held in the memory queue approaches the maximum allowable, they will be deleted if marked with a positive acknowledgement of transmission or stored in non volatile storage if no such acknowledgement has been received.

3.3 Local Storage of evidential records

The intention is that Evidential Records will only be stored in non volatile storage when normal operation is not possible due to such factors as a communications failure or a Instation failure. The number of Evidential Records held in the memory queue can be up to 200, depending upon record size, and is parameterised.

Whilst an Evidential record is held in ram, transmission may be requested repeatedly. However once transferred to the non volatile store, the record will be deleted after one successful transfer.

The instation may use explicit delete commands to remove a record once it has been successfully received. Such usage may obviate the need for an acknowledgement for Evidential record transfer. In which case the system must be configured to not require acknowledge on ER transfers.

The common identifier comprises a pair of numbers. These are the *event_id* and *file_id*.

The *event_id* is an incrementing number issued to each event on the camera. This number will wrap at `2^32`.

The *file_id*, also an incrementing number, will identify the storage location within the camera (the file name is based on this number). This number will wrap at a number based on storage capacity. Thus for example a system with a nominal 1 Gbyte storage and a maximum event file size of 128k will wrap at about 7500.

3.4 Additional context images

When running on a SPIKE+ platform equipped for FULL VES the VES system has the ability to capture additional context images in addition to the primary overview captured at the same time as the plate event. These context images are captured:

- all before the primary event
- all after the primary event
- or evenly split before and after the primary event

The interval between the context images may be specified. The requested interval is restricted by the context image sample rate. By default the sample rate is set to 2 ie a context image is stored in the camera every 40ms and thus the precision of the context interval is 40ms.

3.5 Trajectory data

The SPIKE camera has the ability to compute and report a trajectory for a plate passing through the field of view of the camera. The trajectory report comprises two coordinate pairs `x1,y1` and `x2,y2` together with a time interval. The first coordinate pair represents the approximate centre location of the plate in the prime read location as reported by the captured image and prime time stamp. The second coordinate pair represents the result of a secondary read operation on the plate. The interval is the time interval between the the prime read and the secondary read. The secondary plate position may occur before or after the prime read both spatially and chronologically.

The trajectory data will only be accurate and meaningful if the camera is configured to generate this data by collecting a number of images as the plate passes through the field of view.
To enable this system configure:

```
capture set list_length n
```

where n is set to the number of images retained for analysis. By default this will be set to 1 . Trajectory data will be based on the hardware plate finder and thus quite approximate. For accurate trajectories on fast traffic somewhere between 5-10 may be a suitable value. For slower traffic 15 may be more appropriate.

Note that there are some restrictions when computing trajectories.

> If a vehicle is stationary no trajectory may be computed. If a vehicle is moving very slowly then there may be insufficient movement to acquire a useful trajectory before a plate report is generated. If very slow traffic is normal at a particular site then increasing `capture` parameter `count` may be beneficial.
>
> If a vehicle is moving very fast then only one usable exposure may be acquired and a trajectory may not be computed.
>
> If a vehicle is moving diagonally such that the plate is only fully in the field for a very short distance then such images that are captured may be quite close together and as a result any trajectory computed may have a short baseline.

Guidance on expected file sizes

The table below shows some estimated file sizes based on a jpg factor of 50 for images captured under overcast conditions. If images are captured under bright sunlight they will contain more detail and hence will have larger files.

transfer mode selected	**description**	**estimated file size**
	patch file only	2k – 4k
	patch plus full IR	8k – 24k
	patch plus overview	30k – 50k
	patch plus full IR plus overview	34k – 60k
	patch with embedded text	4k – 8k
	patch plus overview, both having embedded text	33k – 50k
	Patch, IR, overview and two supporting context images. All images with embedded text.	115k

NB:

1) Bright sunlight may increase the size of the full IR image by about 10k

2) 850 nm illumination may also increase the background detail in the full IR image increasing the size by a further 5k

If a compact flash or rotating hard disk is in use then storage can be increased up to the 2 Gbyte maximum available to a FAT16 file system. In this case the maximum file limit of the FAT16 system in use on the camera, 32k files, must be taken into account as well as the expected file sizes for image events. NB: on a 2Gbyte FAT16 file system, storage allocated to individual files grows in 32kbyte increments.

For efficiency and drive reliability file storage space is pre-allocated for each file. This will happen automatically at system start if the file system does not exist. Or, file storage may be cleared and reallocated by means of the ves init command. To prevent error & mistaken damage to a live system this command requires use of the system security code.

1.2 System throughput estimations

Link bandwidth to host is generally provisioned to cope with approximately twice the worst case traffic loading ie estimate worst case file size and one event per second.

The camera is sending several data streams to the host system. The streams comprises summary records, evidential records, heartbeat records, message port records, diagnostics and exceptions. Under significant traffic load the evidential record will encompass over 90% of the total flow, so in first order estimations for bandwidth management, all the other data can ignored.

Evidential records may be sourced either from live traffic or from data saved to the CF disk during a communications or instation outage. The stream from live traffic is naturally throttled by the traffic flow. The stream from compact flash is throttled to limit total bandwidth demand to about one record per second. The data from this source is forwarded to the host system via a second connection running over a lower priority link. Live traffic will always take precedence.

Additionally, large amounts of data may be sourced from the streaming image interface (viewfinder). This interface also throttles data flow a) to ensure that the communications bandwidth is not swamped and b) to ensure that normal anpr operations are not compromised.

4 Streaming interface

Data is transferred to the instation over TCP/IP socket streaming interface. This section describes transfer via the streaming TCP/IP interface. There are two data streams, The full evidential image records and the briefer summary records. Each of these streams are further subdivided into a live / high priority data stream and a low priority data stream carrying records from offline storage.

4.1 Image record

When an image event occurs, one or more of the images may be saved for transfer. These are the plate patch area, the full image containing the plate patch, the overview image and secondary context images (if appropriate options are enabled). The images are combined into one record to which an HMAC authentication record is appended

The system will attempt to open a connection to the image host . If this fails and buffering is not enabled, then the data will be discarded. If buffering is enabled, then the data is saved to disk for transfer later when the connection can be re-established.

If no records are available for transfer then the connection is held open for `link_timeout` seconds before being closed.

The record format comprises a header in ASCII text followed by the image data evidential record blob (binary large object). The header allows the image data to be uniquely identified, checked for completeness and saved on the host system without an immediate need to decrypt the blob.

The detailed data structures within the record format may be found in an appendix to this document. Records in this text are for illustration only and may not reflect production software. In all cases software developers should refere to production header files.

```
MMMM SSSSS sssss QQQQQQQQ UUUUUUU mmm LLLLLL CCCC idid\n
byte   nonce[32];             // random number (#2)

typedef packed struct ves_image_data
{
.
.
}
VES_DATA;

typedef packed struct ves_image_sizes
{
.
.
} VES_SIZES;

// image data goes here - variable length
// patch, full ir, oview, supporting context images

// followed by padding - ER is padded to multiple of 32 bytes (inc digest)
// first byte of padding will have size of padding - minimum pad size is one byte
// further bytes have pad index inserted
// then SHA1 digest (20 bytes)

// all data is transferred in NETWORK BYTE ORDER

byte ves_pad[];               // data is padded to multiple of 32 bytes (#3)
byte ves_digest[20];
```

Header Field	Description	Notes
`MMMM`	4 character magic number indicating a) start of image event record, b) version of image event data	
`SSSSS`	Site identifier – variable length max 30 characters – formed as concatenation of camera id and lane id (see above)	
`sssss`	8 char hex (0 left padded lower case) Current session number	
`QQQQQ`	8 characters hex (0 left padded lowercase) representation of event sequence number. This number may be used as a reference to the NVT interface to identify the evidential record for transfer.	
`UUUUU`	8 characters hex UTC time of image capture (seconds)	1
`mmm`	3 characters hex UTC time of image capture (ms)	1
`LLLLLL`	6 characters hex, overall length of data blob (inclusive of all bytes. The blob starts with the nonce and terminates with the digest)	
`CCCC`	4 characters hex, crc16 of data blob. This crc allows the receiving process to confirm that the blob has arrived intact without need for immediate decryption & and re-authentication.	
`ididid`	eight char hex (variable length) filename identifier. This identifier is used as a reference to records stored on compact flash.	
`\n`		
	All the above in clear text – all following data may be encrypted.	

1) The time stamp of the image will be the time of capture of the primary overview image if this is present. If no overview is present, then the IR image time will be used.
2) The summary record comprising `ves_image_data` is sent in clear text. To prevent this being used as a tool to attack the encryption of the enforcement record, this later record is prefixed with a 32 byte random number.
3) The AES encryption process benefits from the data block being a multiple of 32 bytes. Padding is inserted to ensure that this is the case. Thus it follows that the authentication block is always the last 20 bytes of the blob. There will always be at least one byte of padding. This byte will contain the pad length. Subsequent bytes are loaded with an incrementing count. Thus if five bytes of padding are required they will have values `5,0,1,2,3`

NB: Any routines parsing the header data should assume that additional fields may be added to the text line either as diagnostic or as extended protocols. Thus the header parsing code should accept the fields specified in the text line, then ignore further fields if found up to the newline terminator.

If acknowledgements are enabled, then the system will expect the record transfer to be acknowledged. Further records will not be transferred on a channel until the record has been acknowledged successfully or until all retry attempts have completed.

All binary data is passed in network byte order

Optionally, the system may be configured to use a unique key pair for each evidential record. If the system is configured in this mode then the magic transferred within the header indicates that this is the case, and a key block prefixes the blob.

The key block comprises:

```
typedef packed struct ves_key_block
{
      byte  nonce[16];
      byte  reserved[16];
      byte  ukey_1[32];
      byte  ukey_2[32];
} VKB;
```

The camera software generates `ukey_1` and `ukey_2`. The key block is encrypted with `key_2` received as part of the session control protocol.

Keys `ukey_1` and `ukey_2` are then used to authenticate and encrypt remaining portion of the ER.

No local CRC or authentication is provided as part of the key block. This is deliberate.

4.2 Summary record

The system may optionally forward a summary record for every recorded event. The summary comprises:

```
MMMM SSSSS ssssss QQQQQQQQ UUUUUUU mmm LLLLLL idididid\n
packed struct ves_image_data
```

Where

Header Field	Description	Notes
`MMMM`	4 character magic number indicating a) start of image event record, b) version of image event data	
`SSSSS`	Site identifier – variable length max 30 characters – formed as concatenation of camera id and lane id (see above)	
`sssss`	8 char hex (0 left padded lower case) Current session number	
`QQQQQ`	8 characters hex (0 left padded lowercase) representation of event sequence number. This number may be used as a reference to the NVT interface to identify the evidential record for transfer.	
`UUUUU`	8 characters hex UTC time of image capture (seconds)	1
`mmm`	3 characters hex UTC time of image capture (ms)	1
`LLLLLL`	6 characters hex, overall length of data portion of summary record	
`ididid`	eight char hex (variable length) filename identifier. This identifier is used as a reference to records stored on compact flash.	
`\n`		

Field `ididid` is required on systems implementing a pull transfer protocol. This field is a local reference to the data and must be returned with any transfer or delete requests. This field may not be generated on systems pushing data to the host.

No authentication or encryption is applied to the summary record. The process responsible for transfer of summary records runs at a higher priority than that responsible for image records. Thus, within the constraints of the TCP/IP stack, summary records may arrive in advance of the associated image records.

If there is a communications failure and summary records cannot be transferred to the host system then the records are saved in local non volatile storage for subsequent forwarding to the host system.

If acknowledgements are enabled, then the system will expect the record transfer to be acknowledged. Further records will not be transferred on a channel until the record has been acknowledged successfully or until all retry attempts have completed.

4.3 Batched Summary records

Some applications - particularly those using communications with limited bandwidth such as GPRS may need a mechanism to transfer the VRN together with the minimum of data necessary to retrieve a particular Evidential Record. The batched summary record process provides this capability by using a reduced Summary Record format and by batching records to make efficient use of the communications bandwidth.

A Summary Record batch may constitute a particular numbr of vehicles (`max_batch_size`) or the vehicles within a particular time frame (`bsr_interval`). The system will prepare package and queue for transfer a batch of summary records whenever the first of these limits is reached.

The camera will retain a number of batches in the queue (`bsr_nlog`) until successful transfer. If a connection can be opened but a transfer of a particular batch is unsuccessful for three retries the batch is assumed corrupt and is discarded. If data cannot be transferred, then when `bsr_nlog` batches exist in the queue the oldest batch will be overwritten.

When security is enabled, the individual records in the batch are encrypted (AES256 in CBC mode with SR key 1 and IV=0) as they are acquired. This is to ensure that only encrypted data is retained within the camera. When a complete batch has been assembled, an `hmac` authentication signature is generated (using SR key 1) and appended and then the whole batch is re-encrypted (AES256 with SR key 2).

The format of the batch comprises:

```
    clear text header indicating camera/session details
    16 byte nonce
    individual VRN records (32 bytes/record)
    variable (12 bytes currently)  padding
    20 bytes authentication code
```

This compares with over 400 bytes per VRN for full Summary Records.

The format of the clear text header is:

```
    mmm SSSS ssss nnnn zzzz cccc ii\n
```

where

mmmm	magic
SSSS	site_id
ssss	session number
nnnn	count of VRN records in batch blob
zzzz	size of blob
cccc	crc16 of encrypted blob
ii	incrementing count - loops at `bsr_nlog` - indicating batch number and used by the ack/nak process as the event sequence number
\n	newline

Each individual VRN within the batch has the form:

```
typedef packed struct
{
      dwordsequence;          // event sequence number
      dwordfile_id;           // file id for er retrieval
      char  plate[12];        // null terminated text plate reading
      byte  confidence;       // confidence represented as a percentage
      byte  class;            // vehicle classification
      byte  status;           // status of system (as in heartbeat)
      dwordgps_secs;          // time stamp seconds
      word  gps_ms;           // time stamp ms
      short diff_ms;          // signed difference between GPS and secondary
times
      byte  pad[1];
} VES_BSR;                    // exactly 32 bytes
```

Note that the batch includes reduced forms of the classification and status fields.

4.4 Acknowledgement

If the system is configured to require an acknowledgements on summary or evidential records, then this will be built using the data in the record header.

The acknowledgement is a single line ASCII text of the form:

```
AAAA,SSSSS,sssss,QQQQQQQQ\n
```

Where:

AAAA	Acknowledgement / negative acknowledgement flag This may carry reason for the nak.	
SSSSS	Site identifier – variable length max 30 characters – formed as concatenation of camera id and lane id (see above)	
Sssss	session number - 8 characters hex	
QQQQQ	8 characters hex (0 left padded lowercase) representation of event sequence number.	
\n	record terminator	

Acknowledgement flags defined are:

A000	record is acknowledged – local data may be discarded
A001	Record was not accepted, please resend.

If no acknowledgement is received in `ack_timeout` ms the system will attempt to resend the data. If after half of `ack_retries` attempts at transfer fails the system will close the current connection, reopen the connection and attempt to resend the data. If this fails the host will be marked as off line. An attempt will then be made to forward the data to the alternate host. If `ack_retries` attempts have been made without success then the data will be discarded or transferred to the file system as appropriate.

If attempts to transfer the image fail with receipt of NAK messages, then after`ack_retries` attempts the record will be assumed to be corrupt and will be discarded.

If ACKs are required, then an ACK must be received before a further data set may be transferred.

NB: it is possible that a record will be transferred successfully to the instation more than once (e.g. if the ack arrives after a retry starts). It is the responsibility of the instation to log and discard duplicate records as appropriate.
It is also possible that a record will be transferred to both the primary and alternate servers. Again, it is the responsibility of the instation to resolve this conflict.

5 NVT channel

In addition to the standard telnet control channel, a specific control channel has been implemented. Like telnet this uses a network virtual terminal interface (NVT). But unlike telnet this channel does no negotiation for echo, delete etc and does not echo any input. The port used may be specified as a configuration parameter. This channel is implemented for machine-machine interfaces, specifically interfaces to instation, DSRC or lane control equipment.
This interface can be configured to accept commands either as ASCII strings or as binary packets.

On systems where the host controller can determine promptly (within approx. 5 minutes) whether a vehicle event is a violator, system efficiency can be improved by transferring only violator events to the host. If running in this mode, then when an image event has occurred the camera will inform the host system by means of the summary record. The image data will then be retained by the camera until the data is either overwritten by newer data or is rendered obsolete through age. This channel will expect to receive data set transfer or delete requests.

This channel is the control channel used to pass request to the camera to transfer records when a “pull” architecture is enabled.

Exact implementation of this interface may depend upon the OEM equipment to which the interface is coupled.

5.1 ASCII interface

5.2

Command	Description	Response

`send sss nnnn idididid`	send record nnnn	`* CMD: OK nnnn`
		`* CMD:ERR nnnn`

`del sss nnnn idididid`	delete record nnnn	`* CMD: OK nnnn`
		`* CMD:ERR nnnn`

`init`	delete all records – reinitialise storage system (NYI)	`* CMD: OK nnnn`
		`* CMD:ERR nnnn`

`hb sss`	heartbeat (NYI)	

5.3 Binary interface

This interface is specifically for machine/machine links. Full description of data structures can be found in file `ves_nvt.h` as an appendix to this document. Records in this text are for illustration only.

The camera listens on port `nvt_port` for an incoming connection. Once a connection is established, the process expects to receive input at least every `nvt_timeout` seconds. If no input is received for this period, the connection is closed and the process listens for a new connection. To maintain an open connection when no files must be transferred the host may send periodic heartbeat messages.

The format of a request is:

```
 typedef packed struct
{
       byte   code[4];              // magic number
       dword  length;               // packet length
       int    seq;                  // packet sequence number
       int    request;              // operation command
       dword  event_id;             // record reference
       dword  file_id;              // record identifier
       dword  spare[4];
       word   crc;                  // crc of packet
}VES_NVT_REQ;
```

The host system will increment sequence number on each packet sent.
The record ref and ident are information fields supplied as part of the summary record.

The camera responds with:

```
typedef packed struct
{
       byte   code[4];              // magic number
       dword  length;               // packet length
       int    seq;                  // packet sequence number (copied from request)
       int    result;               // operation result (success or fail)
       dword  event_id;             // record ref to which this result applies
       dword  file_id;              // record identifier to which this result applies
       word   crc;                  // crc
}VES_NVT_RESULT;
```

Heartbeat packets are of the form:

```
typedef packed struct
{
       byte   code[4];                     // magic number
       dword  length;
       int    seq;
       int    val;                  // will always be VES_NVT_HB
       word   crc;                  // crc
}VES_NVT_HB;
```

All binary data is passed in network byte order

The camera checks magic number and packet length. If either of these appear not to make sense, it is likely that synchronisation has been lost. The connection is closed, and the camera awaits a new connection.
If the crc does not compute correctly, then an error result is returned.

Heartbeat packets are echoed back to the host system.

This interface supports two primary commands. These commands either delete the specified record or forward the record to the host system.

5.3.1 Encryption

NVT records may optionally be encrypted. An encrypted NVT request is simply another NVT request type where the package payload is encryped. When decrypted it forms a standard request type packet.
The key for encryption is created anew by the host system on the iniation of each NVT connection. The new key is passed to the camera after a successful challenge and authentication has been completed. The key is encrypted with the shared key (KEK)

5.3.2 Delete request

On receipt of a delete request the process will search first for the specified record in ram buffers, and if this fails, attempt to access the record in Compact Flash memory. Once found the record is marked for deletion. The process then returns a response message to the host indicating that either the record was found and will be deleted or that the record was not found.

If the record is in the process of transfer, then the record will be deleted after transfer has been successful. Otherwise the record is deleted immediately.

5.3.3 Forward request

On receipt of a forward request the process will search first for the specified record in ram buffers, and if this fails, attempt to access the record in Compact Flash memory. If a record cannot be found a response is sent to the host system indicating that the record cannot be found.

The request to forward a record is further subdivided. Requests may be queued for standard priority transfer or for high priority transfer. Once a record has been found and moved onto the appropriate transfer queue, a response is sent to the host system. This does not indicate to the host system when a transfer has been successfully completed, but only that the record has been found and marked for transfer. Optionally a message on the message channel will indicate successful transfer.

If a record has been successfully transferred from a ram buffer but has not been explicitly marked for deletion, then the record will be automatically deleted on either expiry or if the ram buffer is required for a new record. This means that the record is retained in ram for a period which under normal circumstances could be several minutes. A further request for the record will be honoured.

If neither a transfer request or a delete request is received for a record then on expiry or if the ram buffer is required for a new record the record will be transferred to Compact Flash.

If a record has been successfully transferred from a compact flash file, then the record will be deleted immediately after successful transfer.

If the system is running a pull protocol with transfer acknowledges disabled then the system cannot know that a record has been successfully transferred. Under these circumstance responsibility for delete is transferred to the instation. The instation must issue explicit deletes.

5.3.4 Clear request

On receipt of a Clear request the process will lock the compact flash store and then proceed to delete all records not currently in transmission.

5.3.5 Test request

On receipt of a test request the process will create an entry to generate a test record. This test record will be introduced to the system a t a time specified within the request.
If the camera is restarted before the test record is generated then the test record request will be lost. The test record time is specified in UTC seconds/milliseconds and could in theory be at ant time till the year 2030.

6 Session protocol

See notes in appendix E regarding detailed protocol.
(not applicable to basic VES systems)

The enforcement system in which the camera outstation may operate may not be in force 24 hours a day. Evidential records are only required during an enforcement session. Outside of a session the camera may send summary records for observed vehicles.
Evidential record authentication and encryption keys should be changed frequently. This system requires that new sessions be issued with new authentication and encryption keys.
A session protocol exists to initiated and terminate enforcement sessions. This protocol also includes mechanisms for the outstation to authenticate itself to the instation and for the instation to forward session authentication and encryption keys to the outstation.

At the start of a session the instation sends a “start session” message to the outstation. The start session message includes a start time and session length (in minutes). If the start time is in the past but the session is still valid, then the session will start immediately. If the start time is in the future then the session will be pending and will start when the session start time is reached. Only one session can be pending.

If a current session is active when a new session start message is received then the current session will remain active until either the session terminates normally or until the new session start time is reached.

For a session to start, the camera must be in a fit state to generate evidential records. A configuration self check must pass. No tamper indication may be set. The clock must be synchronised.

Any session request which does not meet these criteria is rejected. The camera will return a session reject message including an indication of the reason for rejection.

The instation can request an outstation to terminate any current session at any time. If communications fail during an enforcement session, the system will continue to capture evidential records and if this is enabled, save the records to CF storage for later transfer. If storage fills, oldest records are overwritten. Capture of evidential records ceases as normal at the end of the current programmed session.

1.3 Challenge / response protocol

Session start/stop messages must be authenticated. Key transfer (part of the session start request) must be authenticated. The session protocol includes a challenge handshake protocol. The instation may require camera authentication at any time. If an authentication request is made and fails during an enforcement session then a tamper event has occurred and the enforcement session is terminated.

Should the authentication sequence fail due to incorrect challenge data, the camera will fail silently. The connection will simply close. No clues as to the reason for failure are passed to the host system.

7 System Security & data Integrity

(Not applicable to basic VES systems)

1.4 Encryption & Authentication

The camera appends an authentication record to the blob. This authentication record is built using the HMAC (*NIST FIPS PUB 198*) standard described in RFC2104. An extract of this reference is attached as an appendix – for information only. The HMAC is built using SHA1 (*NIST FIPS PUB 180-1*) and a 256 bit key.

Optionally the whole data record (except the header) may be encrypted. The encryption cycle is AES (*NIST FIPS PUB 197*) with CBC and 256 bit keys .

The CBC initialisation vector (IV) used for transfer of evidential records is set to all zeros. This is possible because the first 32 bytes of the evidential record are a random number. Thus the IV is effectively an encrypted version of the first 16 bytes.

The shared key used for site authentication and session key transfer is derived from a shared secret and site specific data (site id and camera serial number). The derivation is performed using SHA256 (*NIST FIPS PUB 180-2*) to retain a 256-bit key space.
(SHA256 is not used elsewhere in this interface as it is processor intensive)

Optionally a unique key pair may be generated for each Evidential Record. When the system is configured to do this, then the Evidential Record is prepended with a key block containing these keys. The keyblock is encrypted with the session encryption key.

1.5 Random Number Generation

A number of security and authentication functions require the use of random numbers. The RNG used in the camera is based on *ANSI X19.17*. The initial seed is derived form platform specific data and local time. The RNG key derives entropy from the following sources:

- the vehicle detector counter
- a signal derived from video noise
- tcp/ip packet interval

Entropy and seed data bits are mixed with the MD5 hash algorithm. The encryptor in use for random number generation is AES256 .
NB: the entropy sources are based on the variability inherent in a deployed camera. Entropy will be significantly reduced in a test situation where a camera is observing a static plate and being triggered regularly. Entropy is however saved between sessions and system restarts. See appendix for statistical analysis of RNG output.

7.1 Key exchange

There is a requirement to provide security for initial key exchange at session start. The key for this transfer is derived from the secure shared secret.

7.2 Outstation authentication

In an idle state, outside a enforcement session the outstation may send system diagnostics data and summary data for observed vehicles. These are in clear text. However before transfer of evidential records can begin there is a requirement for the outstation to authenticate itself to the instation. This athentication process may take place at any time and within reason, as often as required, but must take place as part of the session start protocol.

7.3 Shared secret

Authentication requires a shared secret between the instation and outstation. The Session key transfer requires a shared key. This key is derived from the shared secret. The shared secret is never exposed and never transferred across the network.

On deployment each camera must have this shared secret loaded. This secret should be specific to a particular camera site id and serial number. No two cameras and no two deployments of any one camera should ever use the same secret. No secret should ever be knowingly reused. Generation and management of the shared secret is a critical instation responsibility. Distribution of this secret must be performed manually via an audited process.

The shared secret should comprise 48 randomly selected ascii characters to provide a bit depth of 256. A protocol is provided for transfer of the shared secret to the camera. This may only be done via the serial port which is only accessible when the camera is dismounted. The shared secret once loaded does not exist in clear text within the camera and cannot be recovered from the camera. The internal form is an SHA256 hash of the supplied data. See notes in appendix E.

7.4 Outstation integrity

Steps are taken to ensure that the integrity of the outstation is monitored and appropriate action is taken to maintain the security of keys and the integrity of evidential records.

In production code there is no mechanism present for the session keys to be printed or otherwise reported by the outstation. Keys are stored locally in an encrypted key safe. The key safe key is generated from a number of sources unique to the hardware of the individual camera (ie even if the key safe could be copied from the camera it would not be readable on any other system).

If the outstation detects potential tamper then the current session is immediately terminated and the session keys are destroyed. Tamper detection may (optionally) comprise any of the following detected conditions:

- Unauthorised opening of roadside cabinet
- Disconnection of either camera cable
- Loss of power – on restart the keys are destroyed
- System restart, manual or as a result of any software failure
- An operator connection with more than three password failures
- If access control list is enabled, a tcp/ip access which is not validated (hack attempt)
- Loss of Gemini pings

If the system does shut down a session for any reason other than power failure, any data which has not yet been encrypted will be lost. This is likely to amount to no more than two or three records. However all other data records will be forwarded directly or saved to flash if the servers are not contactable. On power failure, all data not in flash memory is lost.

If the outstation detects any situation which *may* compromise an evidential record, then flags are raised in the exception and heartbeat systems. The heartbeat status word forms part of each evidential record so potentially compromised evidential records are individually marked. The areas monitored include:

- Operator on line to telnet port or web page
- Operator monitoring system via client or viewfinder interfaces
- Any change in system configuration. Should such a change occur all subsequent records in that session will have the appropriate flag raised.

7.5 System self check

For enforcement systems there is a requirement to ensure that current system configuration is correct and valid before an enforcement session can start and valid enforcement records may be generated.
In this context, "configuration" refers to the set of files including:

- installed program image
- installed ANPR image
- dynamic configuration file
- access control list

The application layer computes a crc of the active configuration and compares this with both the crc of the copy configuration in flash memory and a reference crc in eeprom. For a valid system all crcs must match. A session start request will be rejected if the crcs do not match.

As documented elsewhere, changing configuration during a session will result in a flag being set within each subsequent violation and summary record within that session.

When a deliberate configuration change has been made, for the system to become valid, the backup copy in flash of the configuration must be updated and the eeprom copy of the crc must be updated. Changing the eeprom copy requires use of the system security code.

When a new enforcement session is started, the session accept message returns a copy of the system configuration crc to the instation.

An independent procedure may be required to supply the updated crc to the instation for external system validity audits in any system where these may be required.

1.6 Access control list

All TCP/IP port connections into the camera may be validated against an access control list. The access control list is specified with:

```
>> sys set access_list filespec
```

If the access control list is specified, then it must be present. This is a critical file. This file is saved to and restored from flash when the `key update` and `key restore` functions are invoked. Loss or deletion of the file will render the system application inaccessible over tcp/ip connections. The format of the file is:

```
* a comment line starts with *
* the next line permits access to all tcp/ip port from the specified address
aaa.bbb.ccc.ddd
* the next line permits access to a specified port from the specified address
```

```
aaa.bbb.ccc.ddd ppp
* the next line permits access to a range of ports from the specified address
aaa.bbb.ccc.ddd ppp-PPP
* this line indicates that connections from the specified class C network are permitted
aaa.bbb.ccc.000
* note that there is no mechanism other than specification of a class A, B or C network
* to specify a range of ip addresses
* The access list may contain up to 32 entries. The access list is scanned first to last.
```

The access list file is read when the first incoming external tcp connection is attempted. Any changes to the access list file will require a system restart to take effect.
Any validation failure will result in:

a. the connection attempt being rejected
b. an exception being raised
c. a tamper event

The access control function is only applied to incoming TCP connections. The camera does not expose any UDP or ICMP interfaces which might affect system configuration. The non TCP protocol ports which are exposed are listed below.

protocol	port	function
UDP	configurable (default 7)	echo server. May be set enabled/disabled
UDP	123	SNTP server The camera may act as an SNTP server. This function may be enable/disabled.
ICMP		The ping responder is always enabled.

There are no defences in the camera against denial of service attacks. This is seen primarily as a network responsibility.

7.6 Port protection

The camera VES interface has a number of port connections. These ports are restricted or protected as follows:

Port	description	note
Serial port	This port appears on one of the rear connectors of the camera. It is >4M above ground so is physically difficult of access. This port provides factory configuration and an engineering interface to the software. In production systems the pins to this connector are not connected. The cable & connector is monitored via other conductors. Disconnection of the cable – required to gain access to this port - will result in a tamper alert.	1
Telnet tcpip port 23	An engineering password is required to access this port. Only one connection at a time may be made to this port. The connection within the cabinet is not encrypted. However connections to the cabinet are protected via the firewall and VPN. A connection to this port may generate an exception and raises the operator online flag in the status word.	2,3,4

Port	**description**	**note**
http tcpip port 80	An engineering password is required to access this port. Multiple connections may be made to this port. The connection within the cabinet is not encrypted. However connections to the cabinet are protected via the firewall and VPN. A connection to this port may generate an exception and raises a flag in the status word. The HTTP service may optionally be disabled in systems where this is perceived as too high a risk.	2,3,4
tcpip port 113	Email reverse authentication. See RFC1413. Operational if email facilities are enabled.	
ICMP ping	Any valid ping message received will be echoed.	
UDP echo user configurable port (default 7)	If enabled, this port will echo back any messages received. By default this facility is not enabled.	
tcpip port 3570	This port provides a monitoring tool connecting to the PIPS client application displaying images and plates from vehicle events. (Incoming data on this port may be able to modify the configuration of the camera. If a connection is made to this port the operator online flag is raised in the status word.)	
tcpip port 3577	A connection made to this port will receive a stream of data comprising operational, diagnostic and error messages. Incoming data on this port is discarded.	
tcpip port viewfinder (port 9000)	This port provides a stream of images for diagnostic and observational purposes. The incoming data stream on this port can make restricted changes to the system configuration (image selection, compression etc). A connection to this port may generate an exception and raises the operator online flag in the status word.	4
tcpip port snapshot (port 9001)	A connection to this port receives a single image from the camera. Incoming data is ignored.	
application key port (default 10003)	This port is the session control port. It runs a private protocol. Authentication is required as a prefix to any operations initiated through this port.	
application trigger port (default 10006)	This port, if enabled accepts trigger messages from associated lane control or DSRC equipment. It runs a private protocol. Any messages not matching the private protocol are discarded. The configuration of the system cannot be changed over this port.	
application nvt port (default 10002)	This port provides application specific control, receiving a restricted set of commands either from locally connected lane control equipment or from instation equipment.	
application message port (default 10004)	A connection to this port receives application progress and informational messages. Incoming data on this port is discarded.	

notes:

a) On trial systems this port would be connected back to the roadside cabinet.

b) An attempt to remove the ethernet connection from the camera to make a connection directly to the camera will result in a tamper alert.

c) References to “cabinet” refer to the roadside enclosure housing the communications infrastructure equipment. Typically this will be fibre or copper connection, leased line DSL or

ADSL. The connection to the communications infrastructure would normally be made via a firewall/router and would provide additional security in the form of a VPN.

d) The network infrastructure and roadside housing is provided by others and is not the subject of this document.

e) If the system communicates over GPRS or other wireless service then the roadside equipment might only consist of a power supply.

f) There is an implicit assumption that the roadside “cabinet” if it exists is separately defended and that a mechanism is in place to indicate to the camera that a tamper has been attempted.

g) A system exception may be raised either on any connection, or just on connections made during an enforcement session.

All listening TCP/IP ports may validate incoming connections against an access control list.

1.7 Passwords

The VES system uses a number of passwords alluded to above.

7.6.1 Primary CLI password

The system may be configured to require a password before any access may be made to the CLI interface. If set, a hash of this password is retained in eeprom. The password must be entered to gain access to the system both in application and eprom modes. The entered password is passed through a one way hash function (`sha1`) before being compared with the stored data. Hashing the password ensures that even if access could be gained to the contents of memory the password could not be recreated by viewing ram content.

This password is of primary importance in maintaining system security. If this password is lost the camera may only be recovered through a procedure executed via the serial port. To maintain security integrity this procedure will result in total loss of all keys (including shared secret and key encryption key), all configuration data and all application program images. Evidential records which have been encrypted and stored in compact flash will not be lost.

This password may be of variable length and must consist entirely of printable characters. It is an operator’s responsibility to issue and enforce a password policy.

7.6.2 Web page password

It is likely that in any secure application the HTML interface and web page controls will be disabled. However should the operator choose to retain the web interface, then a password may be set that must be entered before gaining access to the interface.

7.7 Comments on compliance with PSDB guidelines

The original PSDB guidelines (PSDB 3/96) were prepared nine years ago. Technology has moved forward and procedures mandated within these guidelines may no longer be appropriate.

The original guidelines mandate encryption using 56bit DES with 64 bit CBC. The DES algorithm is no longer seen by the industry as being adequately secure – a dedicated machine could be built to break the cipher in about three hours in the mid nineties. Updates to the guidelines in 2002 suggest use of triple DES. This system proposes use of the stronger AES cipher (*NIST FIPS PUB 197 version ???*) with CBC chaining and 256 bit keys.

The guidelines mandate use of DES-MAC as an authentication code. This interface proposes use of the current industry standard which is HMAC (*NIST FIPS PUB 198*) built using SHA1 (*NIST FIPS PUB 180-1*). Again a 256 bit key will be used.

Separate keys are used for authentication and encryption though if desired, a single key could be used for both operations. This is not recommended.

Optionally unique keys may be created and used to authenticate and encrypt the body of each evidiential record. If this facility is enabled then the unique key pair is encrypted with the session encryption key and prepended to each Evidential Record.

Optionally NVT control messages may be encrypted. If this facility is used, then the initial opening of the NVT channel requires a successful CHAP before a key is passed across the link.

The guidelines mandate use of key transfer using triple encryption DES and a third and fourth key, the KEK (Key Encryption Keys) which are to be manually distributed to outstation sites through an audited process, records of which form part of the evidential chain.

This interface provides for the derivation of a 256 bit KEK from the shared secret in use for authentication. Session keys are transferred by means of this 256 bit KEK and AES256. A protected procedure exists for manual installation of the shared secret from a specific hardware port and only from this port. It is a user responsibility to ensure that the shared secret has a reasonable bit depth. But even if a secret of limited bit depth is loaded, this input data is combined with a (known) seed and turned into a 256 bit object using SHA256 (*NIST FIPS PUB 180-2*) to ensure that a key of adequate apparent bit depth is in use.

There is a requirement to ensure that authentication and encryption are applied to the evidential record as soon as possible after capture. Creation of the authenticated and encrypted evidential record has high priority within the camera/anpr system. Under normal circumstances the evidential record will have been created within three seconds of the capture event. Under peak load this may increase but should never exceed ten seconds. Test software within the system can be used to display a histogram of latency between capture and encryption.

The updated PSDB guidelines mandate that control interfaces be encrypted. Optionally the NVT channel may be encrypted (see above) to meet this requirement. Remote interfaces to the anpr/camera system are not encrypted within the system cabinet. There is an expectation that these interfaces would be served to the outstation secure cabinet site via a VPN fully in accordance with PSDB guidelines. Any detected unauthorised access to the communications cabinet or camera cables will result in immediate termination of the current session and deletion of the current session keys. Any detected unauthorised attempts to communicate with the camera over the network (hack attempts) will result in an exception being raised. The VES guidelines assume that the camera will only use encryption for evidential transfer when a session is active. No evidential records are created outside of a session.

However there is still the possibility that an *authorised* access can compromise the system. Accordingly, any connection to any potential camera/anpr control interface either at any time or at minimum during an enforcement session will raise a system exception. Whilst a control connection is active any evidential records captured will have a marker indicating that they may be compromised. Any system configuration changes made during an active session will raise a flag applied to each subsequent record captured in that session. Thus the instation and back office systems can take responsibility to regrade any captured records which may be considered compromised.

8 Fault Tolerance - Host detection & communications failure

Two host IP addresses may be specified. The VES interface software establishes the presence or otherwise of a host by pinging the host. A host which does not respond to the ping is assumed not to be present. The ping is either ICMP or a UDP echo request. The UDP echo port defaults to port 7 but can be specified. Thus the host systems between them can exert control over which is to accept data from the camera. The ping controls are:

`sys set ping mode`	`n`	Where n=0 for ICMP and n=1 for UDP
`sys set ping port`	`nnnn`	set udp port for use by udp echo requests
`ves set ping interval`	`nnn`	seconds between ping attempts to check whether hosts are online
`ves set ping wait`	`nnn`	maximum number of ms to wait for a ping response before deciding that this ping has failed.
`ping aaa.bbb.ccc.ddd`		CLI test routine to send an ICMP ping to specified host
`ves ping aaa.bbb.ccc.ddd`		CLI test routine to send a ping to the specified host using the configured ping mode & port

NB: When the VES interface pings a host, two ping events will be generated as ping, either over ICMP or UDP protocol, is an unreliable protocol.

If UDP ping (udp echo request) is selected then the udp packet contains useful data in the payload. The payload is an ascii string of the form

```
      aaa.bbb.ccc.ddd aaa.bbb.ccc.ddd sssss eeeee nnnnn SSS BBB V site_id ssssss
xxxxx ii
```

where

Field	Description	notes
`aaa.bbb.ccc.ddd`	ip address of target	
`aaa.bbb.ccc.ddd`	ip address of host	3
`Ssssss`	UTC seconds (hex)	
`Eeeeee`	local elapsed ticks (ms) hex	
`Nnnnn`	ping system layer event counter (hex)	
`SSSS`	unit serial number (dec)	
`BBB`	software build number (dec)	
`V`	character indicating ping source	2
`site_id`	site_id	1
`Sssssss`	local application layer status	1
`Xxxxxxx`	extended local status word	1
`Ii`	ping application layer id counter dec – wraps at 99	

1)The application may generate other pings before opening a connection. These fields may not be present. See note 2

2)There are a number of possible sources of echo request messages both within the VES application layer and other application layers within the camera. The data preceding this letter is generated by the echo request module and will be common to all echo requests. The data subsequent to this flag letter is application layer specific. The VES interface generates two types of echo requests. Type V is generated periodically to establish the state of the host systems. Type O is generated before any connection open attempt. Type O requests contain no additional data.

3)The source of the udp echo request could of course be extracted from the receiving socket. However there may be NAT in place between the camera and the host system. Providing this data allows the echo server to establish both the apparent and actual source address.

4)

If the current host goes offline (fails to respond to the two successive ping requests) any current transactions to that host will attempt to complete. The connection will then be closed. The system will then attempt to open new connections as required to the other host if it is online. Thus a managed instation change over would disable the echo server on one instation then hold the instation active for a period to allow pending events to complete.

On change of host status, all VES specific open connections will be closed and reopened.

The message port and NVT port listen. However these ports will also be closed under the circumstances described above, allowing the appropriate instation to establish new connections.

If the primary host is online and accepting data when the alternate host comes online no action is taken.

If the alternate host is online and accepting data when the primary host comes online, then after any current transaction have completed on any port, the connection is closed and reopened as required to the primary host.

Any change in host status results in the generation of an exception message.

NB: each port on the camera (apart from the web interface) accepts only one incoming connection.

Independently, two hosts may be specified for time synchronisation. The system will always attempt the primary before the secondary.

If an attempt to send data to a host fails (as described elsewhere in this document) then that host is marked as offline and the alternate host will be tried if it is showing as available.
Clarify – retry if host is still online

SNTP behaves in a somewhat different manner. The SNTP process will always try the primary before the secondary specified time server.

A synchronisation problem will only be registered if neither server responds.
Synchronisation failure is only reported if neither server can be contacted for in excess of twenty minutes.

9 Message channel

A message channel is available, providing progress and status messages for the record transfer operations. This message channel appears on a separate configurable TCP/IP port. There will be no effect on system behaviour if the port is left unconnected. Messages will be strictly formatted to facilitate machine reading. Messages may be transferred either in plain text or within an encrypted envelope.

9.1 Encrypted Envelope

The content of encrypted messages is identical to plain text messages described below, but each message is placed in an encrypted envelope.

When the message channel connection is first opened the camera is challenged by the host system (CHAP protocol). If the challenge response satisfies the host that the expected camera is communicating then a unique and ephemeral connection key is transferred using the shared KEK (key encryption key) derived from the shared secret. This connection key is then used to encrypt all subsequent message transfers for the duration of the current connection. The messages are encrypted using AES256.

The message channel will be closed by the camera automatically a) if the connection has been idle for 60 seconds (msg_timeout configurable) or b) if the connection has been active for a period in excess of 60 minutes. This ensures that the host system is forced to periodically create a new connection with an associated new ephemeral key.

The host system may also implement a policy of closing and reopening the connection to initiate use of a new ephemeral key.

The format of the envelope carrying the message is:

```
#defineVES_MSG_MAX            128
typedef p372_packed struct
{
       dword  code;                 // magic number
       dword  length;               // packet length
       int    seq;                  // packet sequence number
       dword  msg_len;
       byte   nonce[16];            //
       byte   msg[VES_MSG_MAX];     // encrypted payload (multiple of 16)
       word   crc;                  // crc of packet
} gcc_packed VES_MSG_PAYLOAD;
```

where **msg** is the plain text message described below. The fields **nonce** then **msg** are encrypted AES256.

9.2 Plain text

Messages will:

be ASCII text
be newline terminated
start with a message specific identifier of the form:

```
Lnnn
```

where L is an upper case letter , one of:

M	progress message
W	warning message
E	error message
P	panic message

nnn is a unique numeric message identifier

e.g.

P123 PANIC @ 0x123abcd widget failed

This is an extensible system. In practise the only messages implemented for this application at this time are:

message	Description	notes
`M000 n keep alive`	Keep alive message transmitted every *timeout* seconds (where *timeout* is the configurable parameter described above). `n` is a decimal numeral incrementing on every transmission	1
`M002 site ssss eeee`	Obsolete message site – site_id ssss - session number eeee – local event number	
`M004 c iiiii uuuuuuuu`	Indicates that tcp/ip transmission of image set for camera `c` identifier `iiiiii` at timestamp `uuuuuuuu` (the hex utc time associated with the image set) is complete	
`M005`	A new session has started	
`M006`	A session has terminated normally	
`M007`	A session has terminated abnormally	
`M008 tbd`	Current server has switched	2
`M014 c uuuuuu sss nnnnn ppppp idididid`	indicates that a trigger event has occurred on camera `c` at time `uuuu` Seconds, `sss` ms with sequence `nnnn` and plate `pppp`	
`E001 nnnn string`	data discarded - Image processing Q full	
`E002 nnnn string`	data discarded - Unable to contact (ping) server	
`E003 nnnn string`	data discarded - Server is visible (ping) but transfer socket is unable to connect	

message	Description	notes
W001 tp vvvvv string	data transferred to disk buffer - Image processing Q full	
W002 tp vvvvv string	data transferred to disk buffer - Unable to contact (ping) fserver	
W003 tp vvvvv string	data transferred to disk buffer - Server is visible (ping) process is unable to connect	
W010 tp rrr	Triggers are being received at a rate which may exceed system specification. Events may not be handled correctly.	
H001 tp sssss bbb f	copy of heartbeat data – see below for explanation. NB: heartbeat must be enabled for this message to be created.	

notes

1) This message may be used by a monitoring system to confirm that the lane is fully operational.
2) An indication that the primary server has switched.

10 Heart beat Channel

1.8 Format of heart beat message

If option `heart_beat` is set to non zero, then, every `heart_beat` seconds the 372 will open a connection to the designated host system and send a heart beat message. The connection will be closed after transmission. The heart beat message by its very existence indicates that the camera is operational and that the network connection is intact. The content of the heartbeat message provides a brief indication of the operational status of the camera. The heartbeat status field is also available within the main image evidential record and within the summary record. Bit fields within this status field indicated any potential weakness in system security which may compromise the integrity of an evidential record (e.g. cabinet door open, engineer connected via telnet, time not synchronised).

iiii,ssssssss,bbb,f\n

where:

`iiii`	station identifier – this is the site identification of the 372
`ssssssss`	station status – a hex number . 0 is OK, any non-zero value may indicate a potential system problem. The exact format of this is described below.
`bbb`	beat number - decimal number. wraps at 999. The receiving system may check sequencing to detect any network or system problems.
`f`	decimal number of failures to send heart beat. This will reset to zero on a successful send. If this is ever non zero it may indicate an intermittent network problem.

e.g.:

```
1234_1,00000000,876,0
```

which would indicate

- station 1234_1
- status of station is 0 which is OK
- 876 heart beat messages have been sent
- no heart beat failures since the last successful send

If variable `not_alive` is set non zero, when the number of sequential failures to transmit the heartbeat reaches this number, then the system will attempt recovery action. The most likely outcome will be a system soft reset.

10.1 Status report

There are two status words, providing 64 bits of status.

Do note however that a bit set may not indicate a true fault. One or more of the following bits might be set indicating internal system status:

bit field	description	notes
0x00000001	last heartbeat connection failed	
0x00000002	Primary host is offline	
0x00000004	Alternate host is offline	
0x00000008	last image transfer reported an error	
0x00000010	time out on vehicle detection for camera 1	1,2
0x00000020	Cabinet door is open - UNAUTHORISED ACCESS	
0x00000040	Summary link to host is closed	1
0x00000080	Image link to host is closed	1
0x00000100	System has performed a soft restart. This is the first successful heartbeat transmission since start-up. This bit would be cleared on subsequent transmissions.	
0x00000200	No evidential record session is active	
0x00000400	Configuration has changed within current session	
0x00000800	An operator is online.	
0x00001000	camera 1 is reporting fault state	3
0x00002000	not used	
0x00004000	not used	
0x00008000	not used	
0x00010000	The system has not yet attempted to lock time over SNTP	
0x00020000	The SNTP server did not respond	
0x00040000	The system was unable to synchronise with the SNTP server	
0x00080000	The system was unable to achieve time lock after 20 synchronisation attempts	4
0x00100000	Data has overflowed internal storage or server cannot be reached. Images are being transferred to local disk.	
0x00200000	Internal disk has images to transfer to host	
0x00400000	Internal disk system has overflowed. Evidential records are being overwritten.	
	.	
0x01000000	Disk system is not ready	
0x02000000	Disk system is not present or has not responded to queries	
0x04000000	A disk operation has returned an error	
0x08000000	SPARE	
0x10000000	SPARE	
0x20000000		
0x40000000	System and reference CRCs did not match at system restart	
0x80000000	System has performed a hard restart subsequent to a power failure or operator intervention. This bit will be cleared on subsequent transmissions	

Notes

1) When traffic is low it may be normal for these warnings to appear.
2) This bit is set when no image capture has taken place on the input channel for a period in excess of `timeout` seconds.

3) This status bit will be set if the camera input is indicating no sync signal.
4) This may be normal for a short while if the system has been out of time sync for a period in excess of a few days, either through network disconnection or because it has been switched off. Time lock should be achieved within two-three polling cycles after start up. If this does not happen then a real fault exists.

An extended status word of 32 bits is defined in this specification.

bit field	**description**	**notes**
`0x0000000f`	reserved - GPS time synch subsystem	
`0x10000000`	session time sync status	1
`0x20000000`	session operator status	1
`0x40000000`	session configuration status	1
`0x80000000`	session tamper status	1

notes

1) These bits are cleared after the end of a session, and set during a session if any of the related alarms are set. The sesion end exception will carry a copy of these bits before the status is cleared. Thus this end of session exception may be used to indicate if any alarms have occurred during the session

11 Trigger Channel

The camera may self trigger, may be triggered from the external trigger input if this is available or may be configured to accept trigger messages over tcp/ip or a serial port connection.

3M does not recommend the use of ethernet triggers except under circumstances where traffic is moving very slowly.

The camera listens for a single connection on a tcp/ip port. The tcp/ip messages will be subject to tcp/ip latencies and may not be appropriate for all circumstances. Where possible PIPS would recommend use of serial messages.

The serial channel configuration is three wire, 19200bps, 8 data, 1 stop, no parity and no flow control. If connected to a PC serial port, a null modem link is required.

The trigger message is expected as an ASCII string of the form:

```
STX n .... \n
```

where:

`STX`	message start character
`n`	Channel to trigger (0-3) (required on p357 implementations)
`....`	trigger message payload (must not contain STX or \n and will be ignored in all but specific OEM systems)
`\n`	message terminator

The overall length of the trigger message must not exceed 64 characters. The data received as the payload of the trigger message is transferred to the host system in both summary and image records. Under normal circumstances the trigger payload content is otherwise ignored. Specific OEM systems may have code to parse this data.

With this format, any damage to a trigger message will result in the loss of one trigger as the protocol will always resynchronise on the STX character.

The trigger will be generated on the *completion* of a valid trigger message, so for optimal latency the trigger message should be short. The timestamp of the trigger will be that of the first character (STX) as far as is possible within the constraints of the camera operating system.

In addition, when `trig timeout` is set to non zero the system sends a serial heart beat to the host system every trig_timeout seconds. The format of the heart beat is:

```
A nn\n
```

where nn is an incrementing two digit decimal number starting at 00 and wrapping to 00 after 99.

This channel might also be used to return small data packages to the host system.

12 Image streaming protocols

Two related protocols are provided for transfer of monitor & diagnostic images to the host system. These protocols are known as viewfinder and snapshot.

12.1 Viewfinder

This protocol provides a stream of images from any exposure entry from either the monochrome IR camera or the colour context camera. Dynamic controls (from the instation) are provided to select image source, image compression factor and image scaling. Internal throttling is provided to protect the camera system from being overloaded by the task of delivering these images. 3M can offer a java applet to interface to this protocol.

The camera listens on a specified port. When a connection is made to the port the camera responds by providing a stream of jpeg images. Each image has an ASCII header of the following form:

```
VF vv c s llll\n{binary stream – jpeg image}
```

where:

VF	record prefix
vv	protocol version (currently 1)
c	camera number (always 0 on P372, P382)
s	exposure sequence number
llll	decimal length of following jpeg image

Controls may be sent to the viewfinder process on the opened port. Controls are of the form:

```
vf set param val param val ....\n
```

where parameters and values are:

camera	n	specify camera to monitor (always 0 on P372 or P3892)
seq	n	Which exposure sequence to monitor. Typically seq 0 will be context and seq 1-3 will be IR
interval	nnnn	Interval between images captured. Minimum is 200 ms for low quality images.
quality	nn	jpeg quality factor (1-99) defaults to 50 which suits most requirements
scale	n	set the image scale 4 – approx 180x120 (default) 2 – approx 360x240 1 – not scaled full field i.e. approx 720x240

12.2 Snapshot

This facility provides a single image per tcp/ip connection. The system is preconfigured to select image source, scaling and compression. Typically the system will be configured to return a high quality ¼ vga (approx 360x240) image from the context camera. Such images could be periodically requested from every site (or a selection of sites) and displayed either on public or private web pages to indicate traffic or camera status at the site. The images are delivered in the same format as for viewfinder above. There is no control cmessage chhannel.

Further Viewfinder and Snapshot configuration control can be found within the CLI and web interface. These controls are documented in the web page help texts.

3M provides demonstration applications which connect to these services. Viewfinder is distributed as part of the standard web interface. Snapshot is a stand alone application which returns images from a defined list of cameras.

13 Diagnostics and Exceptions

The camera VES system includes a module to report detailed system diagnostics and exceptions. routine diagnostics may be reported periodically at a configured period – typically 30 minutes. Exceptions will be reported as soon as may be after the event.

Diagnostic and exception events generate files in the outstation file system. These file will be transferred immediately to the host system if this is possible. If this is not possible then the files are retained until the host system again becomes available or until the files are overwritten by subsequent events. The number of files retained before overwrite is a configurable option.

Diagnostics and exceptions are transferred in chronological order. Note though that this may not be true if the maximum number of retained diagnostics or exceptions is increased whilst any records are pending transmission. However once the current retained set are cleared the system will behave as expected.

Diagnostic and exception messages may optionally be stored and transferred in an encrypted format if created whilst a camera is in session. For this to operate a) the appropriate security options must be set and b) the appropriate session start protocol message must be used to enable transfer of the extra keys required to authenticate and encrypt the diagnostic messages.

13.1 Diagnostic reporting system

Command	Parameter name	value	description	data type	notes
`ves diag show`			show VES diagnostic specific system parameters		
`ves diag set`					
	`port`	`pppp`	VES in-station diagnostic port	decimal integer	
	`period`	`nnn`	Prepare and send regular diagnostics after this period expires. Time period given in minutes.	decimal integer	
	`report`	`n`	Flag to turn diagnostic reporting on or off. 0=off, 1=on	decimal integer	
	`nlog`	`n`	Number of log files kept by system if Instation communications are down	decimal integer	See tags
	`min_sample`	`nn`	Minimum number of samples required before a diagnostic value will be reported (shared by all relevant diagnostic values)	decimal integer	
	`ack_timeout`	`nn`	Number of seconds to wait for a diagnostics acknowledgement	decimal integer	

13.2 Exception reporting system

Command	**Parameter name**	**value**	**description**	**data type**	**notes**
`ves exception show`			show VES exception specific system parameters		
`ves exception set`					
	`port`	`pppp`	VES in-station exception port	decimal integer	
	`period`	`nnn`	Exception data is statistical in nature. This sets the average period. Time period given in minutes.	decimal integer	
	`report`		Flag to turn exception reporting on or off. 0=off, 1=on	decimal integer	
	`nlog`		Number of log files kept by system if Instation communications are down	decimal integer	See tags
	`ack_timeout`	`nn`	Number of seconds to wait for an acknowledgement	decimal integer	
	`min_sample`	`nn`	Minimum number of samples required before an exception value will be reported (shared by all relevant diagnostic values)	decimal integer	
	`max_vcpm`	`nn`	Maximum vehicles count per minute before an exception is generated	decimal integer	
	`max_tooc`	`nn`	not used	decimal integer	
	`min_ttr`	`nnn`	Tag to trigger ratio that will generate an exception	decimal integer	
	`min_mpq`	`nnn`	Mean plate quality (hardware trigger quality) that will generate an exception.	decimal integer	
	`min_mtpv`	`nn`	Minimum mean tags per vehicle which will generate an exception	decimal integer	
	`max_ser`	`nn`	Maximum syntax error ratio before an exception is generated	decimal integer	
	`min_mtc`	`nn`	Minimum mean tag confidence (ANPR read) before an exception is generated	decimal integer	
	`max_char`	`nn`	Minimum character height of plate in pixels before exception generated	decimal integer	
	`min_char`	`nn`	Maximum character height of plate in pixels before exception generated	decimal integer	

Command	Parameter name	value	description	data type	notes
	`max_mvpa`	`nn`	Maximum mean vertical plate angle before an exception is generated	decimal integer	
	`max_mhpa`	`nn`	Maximum mean horizontal plate angle before an exception is generated	decimal integer	
	`max_mti`	`nn`	Maximum mean trajectory radius before an exception is generated	decimal integer	

13.3 Message Descriptions

This section describes the diagnostic and exception messages that use the TCP/IP protocol. The messages relate to time synchronisation monitoring, vehicle plate data, status data and status alarms.
The following requirements apply throughout the message descriptions:

- All data is based on an octet, or byte, which is defined as 8 bits where bit 0 is the least significant bit and bit 7 is the most significant bit.
- Where an application message contains a binary value of a size greater than one byte, the value will be sent in Network byte order i.e. most significant byte first.
- Defined but unused bits in an application message must be set to logic ‘0’.
- Data integrity of application messages relies on the TCP/IP protocol. No other error detection will take place.
- Where application message descriptions contain ASCII strings the text must comply with BS4730.
- The fields of each application message are described in one or more tables where each field has a name, type, size in bytes and a description. More than one table is used for a message where there are repeated parts of a message, or where it is necessary to clarify the message structure. The type of each field is either ‘B’ (binary numeric value), ‘A’ (ASCII text) or ‘H’ (ASCII string presenting a hexadecimal numeric value). Where an ASCII string field has a delimiting character, this is included in the size.
- Where application message descriptions refer to byte positions within an ASCII string the position is numbered from 0 which refers to the byte nearest the start of the message.
- Where application message descriptions contain ASCII strings containing hexadecimal values with a 2 character fractional part the value must be divided by 256 to provide the correct value.
- If the integer (non fractional part) value is 0x007fff then this is a special value indication that the system does not have valid data.
- All number values should be assumed to be signed (twos complement) unless otherwise indicated.
- Where an application message field description refers to a maximum or minimum value or a range of values the value contained in the field will be validated by the receiver. If an invalid value is received the message will be discarded.
- Where application message descriptions contain timestamps, the value is co-ordinated universal time, UTC. This number is an unsigned value.

13.4 Common Fields

These message structures are used in more than one message description.

13.4.1 Magic

The following table describes the Magic field. It indicates the type and version of a message. It is a four byte ASCII string containing a two byte hexadecimal value.

Byte	*Description*
0 to 1	Message Type. Where: 03 indicates a Routine Diagnostics message. 07 indicates an Outstation Exception message. 09 indicates a Camera Exception message. 0F indicates an Acknowledgement message. 10 indicates a Negative Acknowledge
2 to 3	Version.

The Routine Diagnostics, Site Exceptions and Camera Exception messages contain the values of various parameters. Each parameter value is reported in a Statistics Field, which is described in the following table. Each Statistics field is an ASCII string containing a hexadecimal number, delimited by a space.

Field	*Type*	*Minimum Size (bytes)*	*Maximum Size (bytes)*	*Description*
Parameter Indicator	A	2	3	One or more ASCII characters followed by a colon(:).
Parameter Value	H	2	9	An ASCII string containing a hexadecimal value delimited by a space (or, where appropriate, a new-line character).
Parameter value	T	2	12	One or more ASCII characters forming a string and delimited by space

13.5 Statistics Message Header

The following table describes the header for Routine Diagnostics, Site Exceptions and Camera Exception messages. Each field is an ASCII string containing a hexadecimal number, delimited by a space. The header is delimited by a new-line character.

Field	*Type*	*Minimum Size (bytes)*	*Maximum Size (bytes)*	*Description*
Magic	H	5	5	The type and version of the message. See 13.4.1.
Outstation Identifier	H	4	5	Number that uniquely identifies this outstation, in the range 2100 – 4999 (834h – 1387h).
UTC	H	9	9	The number of seconds since 00:00 01/01/1970.

13.6 Routine Diagnostics

Direction: ANPR Outstation to Instation
Description:
This message is sent regularly to the Instation, at the configurable interval that defines the diagnostic period, to report the operational status of the ANPR Outstation. The following table describes the structure of the message. It consists of a header line and an outstation diagnostic line followed by a number of camera diagnostic lines, the number indicated by the Camera Count field of the outstation diagnostic line. Where the ANPR Outstation cannot provide a valid value for a parameter, that parameter is not included in the message.

Field	*Type*	*Minimum Size (bytes)*	*Maximum Size (bytes)*	*Description*
Statistics Message Header	-	18	19	Indicates the Outstation Address, the message timestamp and the type and version of the message. See 13.5.
Outstation Diagnostics	H	4	147	A list of Statistics Fields (see Error: Reference source not found). One field for each parameter being reported (see 13.6.1). Each field is delimited by a space apart from the last field which is delimited by a new line character.
Camera Diagnostics	H	4	764	A list of Statistics Fields (see Error: Reference source not found) for each camera. One field for each parameter being reported (see 13.6.2). Each field is delimited by a space, apart from the last field which is delimited by a new line character (i.e. one line per camera).

13.6.1 Outstation Diagnostics Parameters

The following table describes the outstation diagnostic field parameter indicators.

Parameter	*Indicator*	*Description*
Camera Count	A:	The number of cameras being reported on in the range 0 – 4.
Missed Transmission Count	B:	The number of missed communications for this diagnostic period. Indicates the number of attempts to send diag/excep messages to the host system which have failed for any reason.
Mean Communications Lag	C:	The mean communications lag, in milliseconds, for this diagnostic period, with a two character fractional part. Measured from the SNTP service request. The mean time taken to send a request and get a reply.
Worst Communications Lag	D:	The largest communications lag, in milliseconds, for this diagnostics period. Measured from the SNTP service request. The worst time taken to send a request and get a reply.

Parameter	Indicator	Description
Mean Synchronisation Difference	E:	The mean synchronisation difference, in milliseconds, for this diagnostic period, with a two character fractional part. The SNTP measured difference between local time and server time before a correction is applied.
Worst Synchronisation Difference	F:	The largest synchronisation difference, in milliseconds, for this diagnostics period. The worst case SNTP measured difference between local time and server time before any correction is applied.
Mean Number of SNTP Loops	G:	The mean number of SNTP transactions taken to achieve the required time synchronisation in this diagnostic period, with a two character fractional part. The SNTP system attempts to achieve time lock within a specified window (default 200 ms). As the communication link time is variable and asymmetric, the SNTP software applies a low pass filter to the time correction. Thus a single SNTP request/response may not perform a full time correction. If after one operation (with low pass filter applied) the time is still outside the required window, then a further SNTP request is made. Up to 20 requests in sequence will be made before the system stops making requests for this poll interval.
Time Synchronisation Success Count	H:	The number of successful SNTP time synchronisations during this diagnostic period.
Time Synchronisation Failure Count	I:	The number of unsuccessful synchronisations during this diagnostic period.
Time of Last Successful Time Synchronisation	J:	The UTC timestamp of the last successful synchronisation.
No Communications Response From Host Count	M:	The number of communications errors due to lack of a response from the host during this diagnostic period. This indicates the number of attempts to contact the server (either via ping or via TCP/IP connection open requests) which have failed.
modem failures	N:	Number attempts to initialise and establish communication via the modem during this period. Will be zero if no modem is fitted.
	O:	cabinet door switch state
Evidential records transferred direct	P:	Number of evidential records transferred direct to a host system during this period
Evidential Records Buffered	Q:	Number of evidential records transferred to the non volatile (CF) store during this period
Events Lost / Overwritten	R:	Number of events overwritten during this period (full CF device)
ACK failures	S:	Number of transfer acknowledgements lost during this transfer period
expired records	T:	Number of records deleted in this period because they have aged beyond the permitted limit

Parameter	Indicator	Description
memory usage	U:	Indication of memory usage in processor execution space (free memory on heap)
confidence category	1:	Number of plates read during this diagnostic period which are in confidence category 1
confidence category	2:	Number of plates read during this diagnostic period which are in confidence category 2
confidence category	3:	Number of plates read during this diagnostic period which are in confidence category 3
confidence category	4:	Number of plates read during this diagnostic period which are in confidence category 4

13.6.2 Camera Diagnostics Parameters

The following table describes the camera diagnostic field parameter indicators.

Parameter	Indicator	Description
Camera Index	A:	The camera index being reported in the range 0-3.
Mean Plate Quality	B:	The mean plate quality for this diagnostic period, with a two character fractional part. This indicates the mean plate finder output for valid plate images.
Mean Number of Triggers per Vehicle	C:	The mean number of triggers per vehicle for this diagnostic period, with a two character fractional part. For each vehicle there will be a number of potential plate images which will cause the plate finder to generate a trigger.
Mean Flash Setting	D:	The mean flash setting for this diagnostic period, with a two character fractional part.
Mean Gain Setting	E:	The mean gain setting for this diagnostic period, with a two character fractional part.
Mean Shutter Setting	F:	The mean shutter setting for this diagnostic period, with a two character fractional part.
Mean Vehicle Event Count per Minute	G:	The mean number of vehicle events per minute for this diagnostic period, with a two character fractional part. A vehicle event is defined as a sequence of more than one consecutive plate finder triggers where the best of those plate finder triggers exceeds the bottom threshold.
Syntax Error Ratio	H:	The ratio of plates with syntax errors to total plates for this diagnostic period, with a two character fractional part.
Mean Tag Confidence	I:	The mean plate confidence for this diagnostic period, with a two character fractional part.
Mean Horizontal Plate Angle	J:	The mean plate angle to horizontal for this diagnostic period, with a two character fractional part.
Mean Vertical Plate Angle	K:	The mean plate angle to vertical for this diagnostic period, with a two character fractional part.
Mean Character Height	L:	The mean character height, in pixels, for this diagnostic period, with a two character fractional part.

Parameter	*Indicator*	*Description*
Mean Trajectory Gradient	M:	The mean trajectory angle in degrees, for this diagnostic period, with a two character fractional part. Trajectory is measured *r,th* where *th* is the angle across the field of view and *r* is the perpendicular distance between the trajectory and the centre.
Mean Trajectory Intercept	N:	The mean trajectory radius, in pixels, for this diagnostic period, with a two character fractional part.
Mean threshold setting	O:	The mean threshold setting for this diagnostic period. (NB: no fractional part)

13.7 Outstation Exception

Direction: ANPR Outstation to Instation
Description:
This message is sent to the Instation when the ANPR Outstation detects a change in state of the Outstation Exception Flags (see 13.7.2). The following table describes the structure of the message. It consists of a header line followed by an outstation exception line. Information from the outstation event log is appended to the message. This consists of a block of zero or more lines of text contained within two pairs of percent (%%) characters.

Field	*Type*	*Minimum Size (bytes)*	*Maximum Size (bytes)*	*Description*
Statistics Message Header	-	18	19	Indicates the Outstation Address, the message timestamp and the type and version of the message. See 13.5.
Outstation Exceptions	H	4	5	A list of Statistics Fields (see Error: Reference source not found). One field for each parameter being reported (see 13.7.1). Each field is delimited by a space apart from the last field which is delimited by a new line character.
Log Data Start	A	3	3	A pair of percent characters (%%).
Log Data	A	0	Un-defined	Zero or more lines of ASCII text from the outstation event log.
Log Data End	A	3	3	A pair of percent characters (%%).

13.7.1 Outstation Exception Statistics

The following table describes the outstation exception fields.

Exception Statistic	*Indicator*	*Description*
Outstation Exception Flags	A:	Each bit indicates the state of the Outstation Exception Statistics as described in Outstation Exception Flags (See 13.7.2).

13.7.2 Outstation Exception Flags

The following table describes the bit positions of outstation exception flags.

Exception Statistic	*Bit Position*	*notes*
OS Fatal Error	0	2
FPGA Load Failure	1	2
Hardware Watchdog Reboot	2	2
Soft reset (software watchdog reboot)	3	2
Power Up or Hard reset	4	2
Engineer Connected – telnet/web	5	1
Engineer Connected – client/viewfinder	6	1
Session Start	7	2
Session End	8	2
Clock Drift	9	1
Remote server loss/recovery	10	1
Possible Disk (Compact Flash) problem	11	3
spare	12	
Remote network failure	13	1
Local network or cable failure (tamper)	14	2
Unauthorised cabinet door open (tamper)	15	2
Network tamper/hack	16	2
Password failure	17	2
config change	18	
time sync lost (no sync for over 20 minutes)	19	1

notes

1) this exception is set and an exception message created when the state arises. A further exception message is generated when the state clears

2) this exception gives rise to a single exception event with the specified bit set.

3) Once set, this bit will remain set until system restart

13.7.3 Outstation Exception Data

The following table describes the camera exception fields.

Parameter	*Indicator*	*Description*
tamper source data	O:	bit 0 indicates the state of the tamper switch – 0 indicates OK, 1 indicates tamper switch set
system status word	S:	This is the system status word as reported on heartbeat and in each ER
Current session number	T:	This is the current session number or zero if no session active. Will also be zero on systems which do not support sessions
extended status word	U:	this is the extended status word as reported in heartbeat and in each ER
lost exception count	V:	indicates the number of exception messages lost in this session. Cleared to zero at the start/end of a session.

13.8 Camera Exception

Direction: ANPR Outstation to Instation

Description:

This message is sent to the Instation when the ANPR Outstation detects a change in state of the Camera Exception Flags associated with a particular camera (see Error: Reference source not found). The ANPR Outstation re-evaluates the Camera Exception Flags after the configurable interval that defines the exception period. The following table describes the structure of the message. It consists of a header line followed by a camera exception line. Information from the outstation event log is appended to the message. This consists of a block of zero or more lines of text contained within two pairs of percent (%%) characters.

Field	*Type*	*Minimum Size (bytes)*	*Maximum Size (bytes)*	*Description*
Statistics Message Header	-	18	19	Indicates the Outstation Address, the message timestamp and the type and version of the message. See 13.5.
Camera Exceptions	H	8	158	A list of Statistics. One field for each parameter being reported (see 13.8.1). Each field is delimited by a space apart from the last field which is delimited by a new line character.
Log Data Start	A	3	3	A pair of percent characters (%%).
Log Data	A	0	Un-defined	Zero or more lines of ASCII text from the outstation event log.
Log Data End	A	3	3	A pair of percent characters (%%).

13.8.1 Camera Exception Statistics

The following table describes the camera exception fields.

Parameter	*Indicator*	*Description*
Camera Index	A:	The camera index being reported in the range 0-3.
Exception Flags	B:	Each bit indicates the state of the Camera Exception Statistics as described in Camera Exception Flags (See Error: Reference source not found).
Vehicle Event Count per Minute	C:	The number of vehicle events per minute for this exception period, with a two character fractional part. A vehicle event is defined as a sequence of more than one consecutive plate finder triggers where the best of those plate finder triggers exceeds the bottom threshold.
Tag to Trigger Ratio	E:	The ratio of recognised plates to potential vehicles for this exception period, with a two character fractional part.
Mean Plate Quality	F:	The mean plate quality for this exception period, with a two character fractional part.
Mean Number of Triggers per Vehicle	G:	The mean number of triggers per vehicle for this exception period, with a two character fractional part.
Syntax Error Ratio	H:	The ratio of plates with syntax errors to total plates for this exception period, with a two character fractional part.
Mean Tag Confidence	I:	The mean plate confidence for this exception period, with a two character fractional part.

Parameter	*Indicator*	*Description*
Mean Character Height	J:	The mean character height, in pixels, for this exception period, with a two character fractional part.
Mean Character Height	K:	The mean character height, in pixels, for this exception period, with a two character fractional part. (For compatibility reasons this value is identical to J above.
Mean Horizontal Plate Angle	L:	The mean plate angle to horizontal for this exception period, with a two character fractional part.
Mean Vertical Plate Angle	M:	The mean plate angle to vertical for this exception period, with a two character fractional part.
Mean Trajectory radius	N:	The mean trajectory radius, in pixels, for this exception period, with a two character fractional part.
Expired records deleted	T:	The number of records found to be older than the permitted maximum age during this exception period.

13.9 Camera Exception Flags

The following table describes the bit positions of camera exception flags.

Exception Statistic	*Bit Position*
Vehicles events per minute	0
not used	1
Tag to Trigger Ratio	2
Mean Plate Quality	3
Mean Triggers per Vehicle	4
Syntax Error Ratio	5
Mean Tag Confidence	6
Maximum Character Height (Mean has exceeded maximum threshold)	7
Minimum Character Height (Mean has exceeded minimum threshold)	8
Mean Vertical Plate Angle	9
Mean Horizontal Plate Angle	10
Mean Trajectory Radius	11
Loss of Video	13
Expired records deleted (set if any records in this period)	14

All of the above exception bits are set when the problem arises, and cleared when the problem clear**s.**

13.9.1 Diagnostic & Exception acknowledgement

This message is sent to the ANPR Outstation as the response to a Diagnostic or exception message. The following table describes the structure of the message. It consists of a line including the Magic, UTC, and Sequence Number fields as space delimited ASCII strings containing hexadecimal numbers.

Field	*Type*	*Minimum Size (bytes)*	*Maximum Size (bytes)*	*Description*
Magic	H	5	5	The type and version of the message. See 13.4.1.
UTC	H	9	9	The UTC value corresponding to the time the message was transmitted.
Sequence Number	H	2	5	The number of Ack/Nak messages transmitted by the Instation (wraps to 0 after FFFFh).

example ack message

```
0F01 4587b10b 03
```

14 Dummy record generation

A tool is provided to generate and trace one or more dummy records through the camera software.

Command		*Description*
Ves test new		Insert a new dummy test record
Ves test show		Display the status of the last ten dummy records generated
Ves test clear		Clear the dummy record trace history

Examples

```
>>vest test clear
dummy trace records cleared
* CMD:OK
>>
>>ves test new
issue dummy event
dummy: slot 0 allocated, file id: 0, event id: 47987
* CMD:OK
>>
>>ves test show
dummy trace: 1 records
created                    seq        sr_tx  er_req er_tx  er_del
Fri May 19 17:06:34 2006, 00047987    000    ---    000    ---
* CMD:OK
>>
```

This last display shows the time of creation and the delay to transmission of summary record, the evidential record request (pull mode only), the evidential record transmission and the evidential record delete request (pull mode only).

Test events may also be initiated via the NVT control channel. Using this channel a pending test event may be created which will automatically be inserted into the system at a predetermined UTC time. Note though that when a camera is restarted any pending test events are lost.

15 References

15.1 Internal References

(3M documents)

Title	Description	notes
ANPR Camera user manual.	The primary user manual for the respective Spike(+), SpikeHD or Spikelet ANPR camera.	
ves_data.h	Description of summary and evidential record data structure	
ves_nvt.h	Description of binary control interface data structures	
ves_session.h	Description of session control interface data structures	

15.2 External References

Title	Description	notes
NIST FIPS PUB 198	Description of the hmac authentication standard.	
RFC2104	This document describes HMAC, a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, e.g., MD5, SHA-1, in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.	
NIST FIPS PUB 180-1	Description of the sha1 secure hash algorithm	
NIST FIPS PUB 197	Description of the Advanced Encryption Standard	
NIST FIPS PUB 180-2	Description of the sha256 secure hash algorithm	
ANSI X19.17	Industry standard Cryptographic Random Number Generator	
RFC1321	MD5	
PSDB 3/96	Outline requirements and specifications for Automatic Traffic Enforcement Systems	1
PSDB 13/97	The Speedmeter Handbook	1
PSDB 07/2002	Update to PSDB 03/96 provided privately as part of the VES development process.	1
PSDB Digital Imaging Procedure (Mar/2002)		1
Schneier 96	Applied Cryptography – Bruce Schneier 1996	
Schneier 03	Practical cryptography – Niels Ferguson & Brice Schneier 2003	

notes:

1) The Police Scientific Development Branch, now known as the Home Office Scientific Development Branch (http://scienceandresearch.homeoffice.gov.uk) publishes guidelines for the UK police force:
"T*he HOSDB aims to be the Home Office's definitive source of advice on scientific and technical issues, to deliver innovative technical capabilities and to support our customers, ensuring scientific and technical solutions are effective on the ground*"

This guidance covers, among many other topics, procedures and standards on ANPR systems and Digital Imaging which if followed should ensure that evidence gathered will be suitable for production in a criminal trial.

These documents and others issued by PSDB have been significant influences in the design of this Violation Enforcement System.

15.3 Glossary

The following abbreviations are used in this document:

abv.	***expansion***
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
ANPR Processor	ANPR Processor function within the Spike+ unit
ANSI	American National Standards Institute
CBC	Cipher Block Chain
CCS	Camera Configuration System
CF	Compact Flash
CLI	Command Line Interface
CRC	Cyclic Redundancy Check
DES	Defence Encryption Standard
DR	Disaster Recovery
DSRC	Dedicated Short Range Communication
ER	Evidential Record
EWA	Engineers Workstation Application
FAT	File Allocation Table
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
HFOV	Horizontal Field Of View
HMAC	Hash function Message Authentication Code
HOSDB	(UK) Home Office Scientific Development Branch
IMEI	International Mobile Equipment Identifier
IP	Internet Protocol
IR	Infra Red
KEK	Key Encryption Key
MAC	Message Authentication Code
NIST	(USA) National Institute of Standards & Technology
NVT	Network Virtual Terminal
NYI	Not Yet Implemented
OEM	Original Equipment Manufacturer
PSDB	(UK) Police Scientific Development Branch
RADIUS	Remote Authentication Dial In User Service
RFC	Request For Comment

abv.	***expansion***
RNG	Random Number Generator
SIM	Subscriber Identity Module
SNTP	Simple Network Time Protocol
SRAM	Static Random Access Memory
TBD	To Be Defined
UDP/IP	User Datagram Protocol/Internet Protocol
UTC	Co-ordinated Universal Time
VES	Violation Enforcement System
VPN	Virtual Private Network
VRM	Vehicle Registration Mark

16 Appendices

16.1 Appendix A – Example Initial system configuration

This section describes the changes which may be required to configure a shipped P372 unit for a specific customer site. Appended to this is an example configuration file extracted from a test unit at our offices.

This section contains the default configuration loaded into each camera during manufacture. This is an example file. Actual settings for a specific camera system may vary.

```
*
* ANPR configuration script
*
* default delivery configuration for TFL build 151
*

*set server 192.168.2.6
ftp user ftp_boot
ftp password ftp_boot


set config mem:/system.ini

* P372 Camera Configuration Table
* camera 1
set cam 1 e:1 p:2 v:1 s:0 g:4 w:260 t:100
set cam 1 m:1

* camera 2
set cam 2 e:1 p:1 v:0 s:8 g:3 w:200 t:5
set cam 2 e:2 p:1 v:0 s:10 g:3 w:200 t:5
set cam 2 e:3 p:1 v:0 s:11 g:4 w:200 t:5
set cam 2 m:3

set mode 0x00
set sync 625

* system:
system set flex flash;3722acyc.x00
system set exposure mem:/expose.cnf
system set startup mem:/startup.scr
system set time_server 10.20.100.4
system set alt_time_server 10.20.100.4
system set font8 flash;font_8.8k
system set font16 flash;font_16.32k
system set route ves
system set access_list mem:/access.txt
system set daytime_port 0
system set time_zone 0
system set time_poll 300
system set sntp_latency 1000
system set sntp_window 200
system set sntp_debug 0
system set sntp_max 0
system set brownout 125
system set powerdown 5000
system set idle_time 0
system set idle_mode 0x7
system set plate_type 0
system set plate_max 120
system set plate_min 50
system set t_enable 0
system set t_period 600
system set ftp_debug 0
system set tn_timeout 600
system set cc_eds 0
system set reload 0
system set nmea 0
system set ping_mode 1
system set ping_port 10010
system set sysdump 0

* active:
active set days
active set start_1
active set end_1
active set start_2
active set end_2
active set enable 0
active set debug 0

* client:
client set patch 1
client set sum 0
client set debug 0
client set config 1
client set threshold 50

* vf:
vf set camera 0
vf set seq 0
vf set interval 200
vf set port 9000
vf set debug 0
vf set ratio 0
vf set quality 50
vf set scale 4
vf set persistence 0
vf set snap_cam 0
vf set snap_seq 0
vf set snap_qual 50

* jpeg:
jpeg set host 10.20.100.6
jpeg set account jpg_test
jpeg set password jpg_test
jpeg set separator ,
jpeg set list 0
jpeg set time 60
jpeg set patch 1
jpeg set dir_size 100
jpeg set debug 0
jpeg set name 0
jpeg set quality 50
jpeg set applet 0
jpeg set box 0
jpeg set aspect 0
```

```
* bmp:
bmp set host 10.1.1.1
bmp set account bmp_test
bmp set password bmp_test
bmp set separator ,
bmp set list 0
bmp set time 60
bmp set patch 1
bmp set dir_size 100
bmp set info 0
bmp set debug 0
bmp set name 0
bmp set flags 0
bmp set box 0
bmp set threshold 0

* ves:
ves set host 10.20.100.6
ves set alt_host 10.20.100.6
ves set name LANE
ves set lane 1
ves set device dos:
ves set directory /ves
ves set cabinet
ves set router
ves set security 2
ves set crypt 2
ves set session_opts 2
ves set key_port 10003
ves set image_port 10000
ves set image_timeout 5
ves set link_timeout 20
ves set summary_port 10001
ves set nvt_port 10002
ves set nvt_timeout 45
ves set nvt_mode 0
ves set trig_port 10006
ves set transfer 0x816f
ves set timeout 300
ves set debug 0x0000
ves set ping_interval 30
ves set ping_wait 100
ves set nlog 7500
ves set file_size 128
ves set ack_enable 3
ves set ack_retry 3
ves set ack_timeout 500
ves set scan_interval 30
ves set threshold 35
ves set orientation 2
ves set rules 0
ves set match_window 700
ves set jpeg_size 22
ves set context_count 2
ves set context_offset 240

* anpr:
anpr set file flash;uktcc.eng
anpr set debug 0
anpr set retry 40
anpr set retry_oview 0
anpr set retry_thresh 180
anpr set enable 1
anpr set upper_limit 0
anpr set ir_plate 1
anpr set detect 7
anpr set ap_debug 0
anpr set plate_shape 1
anpr set whiteonblack 0
anpr set multiple 0
anpr set roi 1
anpr set hazard 0
anpr set part_load 70
anpr set full_load 100
anpr set part_queue 10
anpr set full_queue 20

* log:
log set host 10.20.100.6
log set account tfl_test
log set password tfl_test
log set path ./
log set device mem:/
log set size 100000
log set mode 0
log set nzip 1
log set format 1
log set event_level 6
log set debug 0

* pdb:
pdb set file mem:\plates.db
pdb set separator ,
pdb set host 10.1.1.1
pdb set account wl_test
pdb set password wl_test
pdb set update update.csv
pdb set enable 0
pdb set debug 0
pdb set hashsize 50021
pdb set threshold 75
pdb set update_time 0

* capture:
capture set filter 0x0500
capture set raw_mask 0
capture set time 1600
capture set horizontal 50
capture set vertical 50
capture set queue 50
capture set enable 0x0001
capture set blanks 10
capture set count 75
capture set upper_limit 48
capture set lower_limit 0
capture set left_mask 0
capture set right_mask 0
capture set debug 0
capture set age 0
capture set partial 0
capture set duplicates 2
capture set duplicate_level 80
capture set duplicate_depth 10
capture set duplicate_age 30
capture set closeloop 0
capture set lo_thresh 0
capture set age_bias 0
capture set direction 0
capture set close_oview 1
capture set dma_mode 0
capture set long_images 0
capture set conf_shift 0
capture set noise_thresh 30
capture set noise_dip 25
capture set reload_flex 0
capture set reload_flex_activity 0
capture set ctx_sample 2

* closeloop:
closeloop set debug 0
closeloop set threshold_on 1
closeloop set badread_on 1
closeloop set brightness_on 1
closeloop set width_on 0
closeloop set bright_mask 0x000F
closeloop set bright_cutoff 180
closeloop set bright_overexp 200
closeloop set bright_def 5
closeloop set bright_scene 130

* trigger:
trigger set mode 0
trigger set delay 0
trigger set distance 100
trigger set speed 50
trigger set open 250
trigger set window 700
trigger set debug 0
trigger set units 0
trigger set period 2000
trigger set mask 0
trigger set fast_debounce 0
trigger set polled 0

* html:
html set home flash;html.z20
html set default
html set user
html set password
html set enable 1
html set debug 0
html set idletimeout 20
html set max_cache 86400

* mbip:
mbip set host
mbip set lo_script night.txt
mbip set hi_script day.txt
mbip set sensor 0
mbip set enable 2
* mbip set threshold 348
mbip set h_time 10
mbip set h_level 34
mbip set debug 0

* mail:
```

```
mail set host 10.1.1.1
mail set addressee gaw
mail set sender gaw357
mail set userid gaw
mail set timeout 4000
mail set retries 6
mail set delay 24
mail set debug 1

* net:
net set mask 255.255.0.0
net set bcast 192.168.255.255
*net set gateway 192.168.2.1
net set script mem:/net01.scr


* kermit:
kermit set device dosfile
kermit set debug 0

* alert:
alert set site_name site_0
alert set keep_alive 09:00:00
alert set sms_0
alert set sms_1
alert set sms_2
alert set sms_3
alert set sms_4
alert set timeout
alert set debug 0

* action:
action set debug 0

set route ves
```

16.2 Appendix B - Example INSTATION server

As a means of testing the link from the camera to the interface a simple server program has been generated to test the features of a client camera. . This section describes this server software which is provided as part of the camera delivery in order to facilitate set up and test of cameras

. Please refer to the limitations of the scope of delivery outlined in the footnote below.[1]

16.2.1 Name

```
ves.exe
ves_session.exe
ves_trig.exe
ves_monitor.exe
```

16.2.2 Synopsis

```
ves [options]
```

where options are:

-m nnnn	message port on 372	9004
-h nnnn	port listening for heartbeat connections	9005
-c nn	Set connection timeout (seconds)	15
-i nnnn	port listening for image / evidential record connections	1000
-s nnnn	port listening for summary record connections	9001
-k nnnn	port on 372 listening for session control and key transfer operations	9003
-d nnnn	port listening for diagnostic messages	9006
-e nnnn	port listening for exception messages	9007
-t nnnn	Specify image transfer timeout (ms)	5000
-p directory	specify parent home directory for log files	
-n nnnn	port on 372 listening for nvt control	
-D 0xnnnn	debug bit field	0xffff
-C hhhh	Number of hours data to be retained. Data older than this will be deleted from the system, In addition, at system startup, data for the previous ten days will be deleted if found. Set number of hours to zero to disable this facility.	96

[1] *This software is offered as part of the camera delivery. This software has not been through an exhaustive field test process. Due to the experimental state of this software, 3M makes no representations regarding its use, or performance. 3M accepts no responsibility for any expenses, losses, or action incurred or undertaken by any party as a result of the use of this software. This software is provided "AS IS" and "WITH ALL FAULTS". 3M provides no warranties either expressed or implied in law, including IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. In no event shall 3M have any liability whatsoever for incidental and consequential damages as a result of the performance, use or operation of this software. The customer shall have the sole responsibility for adequate protection or back-up of systems or data used in conjunction with this software.*

`-P n`	select protocol 1 – PUSH (default) 2 - PULL	
`-S 0xff`	bit field to specify which records are forwarded to SQL database (if this is supported) 1 – send summaries 2 – send Ers 4 – send diagnostics 8 – send exceptions	`If compiled in, then default is 0x03`
`-T filespec`	Specify html template file	
`-j n`	Enable (1) or disable(0) dismantling of ilf file into separate jpegs	`1`

```
ves_session  [options] {start|stop}
```

where options are:

`-k`	port on 372 awaiting session control & key exchange messages	
`-l nnnn`	session length (minutes)	
`-p dirspec`	specify parent home directory (where camera list will be found)	
`-t nnnn`	offset in seconds to start time of session – may be negative	
`-D 0xnnnn`		

```
ves_trig [options]
```

where options are:

`-D 0xnnnn`	set debug bit field	`0xffff`
`-t nnnn`	trigger port on 372	`9006`
`-p nn`	pass count – number of times to generate the pattern described in the file.	`1`
`-f filespec`	name of file with trigger pattern	`pass_file.txt`

```
ves_monitor [options]
```

where options are:

`-a “prog options”`	define program to monitor	***ves***
`-d 0xnnnn`	set debug bits	0
`-i nnnn`	interval (seconds) between echo requests	5
`-p nnnn`	port to which echo requests should be sent	10010

16.2.3 Description

These programs are simple applications written to compile with GCC and run in a UNIX or LINUX environment. They are supplied compiled using the CYGWIN libraries to run under an NT4 or WIN2K system.

The programs use the posix thread library and where appropriate create a new thread for each connection received. Thus these programs can manage several camera units simultaneously.

The program `ves_monitor` will start the server package, and then restart the server should it subsequently fail. It operates by periodically sending a UDP echo request to the server package. The assumption is that if the server does not return the echo then it has failed.

The program `ves.exe` is a server package accepting connections for data sourced at any cameras which recognise the host system as a valid server.

Summary records are received, reported and logged to summary.log. Evidential records are received and saved to a time stamped directory. The records are dismantled and the images are also saved to the time stamped directory. An html file is created which allows the user to view or print the data set for an event. For the convenience of familiarity and easy comparison this is presented in roughly the same format as a VES penalty notice. A parent directory is created for each day. Within that directory a new subdirectory is created for each ten minute period. The times used to name the directory and files are taken from the event timestamp within the record.

Summary events, image events, messages, heartbeats, diagnostics and exceptions are each logged to separate files within the parent daily directory for each camera. Timestamps within these log files are PC local UTC time.

The program scans file `cameras.txt` and attempt to open a message port connection for each camera description found in the file. See below for the file format.

Heartbeat and message port messages received are printed on the screen and transferred to appropriate log files. Apart from logging the data, no other action is taken.

Terminate the program with ^C.

The program `ves_session.exe` provides a facility to start and stop evidential sessions. On start up the program scans the file `cameras.txt` and for each camera in the file initiates a new session with the outstation. The session number is taken from a file `session.txt`. The number is incremented and written back to the file. Thus subsequent sessions will have incrementing session ids. Site ids and shared secrets are also taken from the file `cameras.txt`. The keys transferred to the outstation are taken from files `key1.txt` and `key2.txt`.

This is a test application, not a deliverable product. No action is taken to maintain any security at this end of the connection. No attempt is made to run in an efficient manner. No analysis has been performed as to the actual extensibility and scalability of this software.

The program `ves_trig.exe` provides a facility to send a pattern of triggers to each outstation for system load simulations.

The trigger control file has the form:

```
*
* ves trigger  test file
*
* format is
* interval,payload
*
```

```
* no payload, no trigger, just a delay
* payload is string with no commas
*
100

3000,00
3000,example_payload
3000,2
3000,3
3000,4
3000,5
3000,6
3000,7
3000,8
```

Where lines starting with * or blank are ignored. The first field is an interval in ms. If there is no 2nd field then only the delay specified in the first field is executed.

The trigger message sent to the camera is of the form:

```
STX ssssssss mmm oem_data\n
```

where:

STX	trigger record start marker
ssssssss	eight digits hex UTC seconds
mmm	three digits hex UTC ms
oem_data	Oem data field taken from 2nd field in file. This oem data field should not contain comma characters
\n	message terminator

The overall length of the trigger message must not exceed 64 characters. This restricts the size of the oem data field in this test program.

16.2.4 Installation

Copy the programs `ves.exe, ves_session.exe, ves_trig.exe` and the dll `cygwin1.dll` into a directory somewhere. Create two key files `key1.txt` and `key2.txt`. Either execute from a command prompt directly or via a desktop shortcut or by browsing and clicking on the program name. The program could be executed by any startup script if required.

The list of camera outstations is in file cameras.txt. The format of the file is multiple lines of the form:

```
aaa.bbb.ccc.ddd,site_id,secret,filespec
```

where

aaa.bbb.ccc.ddd	ip address of outstation/camera

site_id	site id of camera. This must be accurate as it is required information for the authentication process.
secret	This is the secret shared between the camera and instation for the purposes of authentication and key exchange. May not include comma.
filespec	This is the filename used for storage of snapshot files.

Blank lines, and lines starting with * are ignored.

16.2.5 Example output

```
VES violation enforcement system
Instation simulation
built Feb 18 2005 08:46:40
      P357 ip address: 62.49.172.84
      heart beat port: 9005
         message port: 9004
         summary port: 9001
           Image port: 10000
             Key port: 9003
       home directory: ./
       security level: 2
key_length: 256
ves: udp echo server starting
ves: diag server starting
ves: excep server starting
ves: heartbeat server starting
ves: message monitor starting
ves: image server starting
ves: summary server starting
ves_msg: connected
ves_session: starting
ves_session: connected
ves_msg: M004 lane_1 00000004 4215a931
ves_msg: M000 7784 keep alive
ves_summary: accepted fd:10, thread 168192568
ves_summary(51): TS01 lane_1 00000004 0000000b 4215a959 0f1 000000b8
ves_summary: expecting 184 of 184
ves_image: accepted fd:13, thread 168192752
ves_image(54): TI01 lane_1 00000004 0000000b 4215a959 0f1 008100 8e9a
ves_image: expecting 33024 of 33024
ves_image: decrypt with key >0001020304050607000000000000000000000000000000000
0000000000000<: 4213441
ves_image: decryption: 3mS
ves_image: authenticating with 256 bits of  key >00010203040506070000000000000
000000000000000000000000000000000<
ves_image: authentication OK in 1mS
patch size:  3341
 full size:  4891
oview size:  8178
ves_msg: M002 lane_1 00000004 0000000b
```

16.3 Appendix C – Analysis of random number generation

Security of the system is dependant on (among many other considerations) the quality of the random number generator. A statistical package and data collection tools are available to analyse the output of the RNG in the camera.

The statistical analysis program and documentation may be found on:
`http://www.fourmilab.ch/random/`

(If this cannot be located a copy may be obtained from 3M)
Data suitable for this program may be collected on the camera by issuing the command:

```
      >> encrypt rand nnnn filespec interval
```

where

nnnn	number of random numbers to collect
filespec	file in which to collect the numbers
interval	Interval seconds between collection of random numbers. This should be comparable to the vehicle interval experienced whilst data is being collected.

The result is a random bit/byte stream in filespec.
Data may be analysed either as a series of bytes or as a series of bits.

example:

```
>>encrypt rand 5000 rand2.dat 5
* CMD:OK
>> ftp put rand2.dat
```

```
> ent rand2.dat
Entropy = 7.997809 bits per byte.

Optimum compression would reduce the size
of this 80000 byte file by 0 percent.

Chi square distribution for 80000 samples is 242.09, and randomly
would exceed this value 50.00 percent of the times.

Arithmetic mean value of data bytes is 126.8848 (127.5 = random).
Monte Carlo value for Pi is 3.164479112 (error 0.73 percent).
Serial correlation coefficient is -0.006526 (totally uncorrelated = 0.0).

> ent -b rand2.dat

Entropy = 0.999992 bits per bit.

Optimum compression would reduce the size
of this 640000 bit file by 0 percent.

Chi square distribution for 640000 samples is 7.01, and randomly
would exceed this value 0.50 percent of the times.

Arithmetic mean value of data bits is 0.4983 (0.5 = random).
Monte Carlo value for Pi is 3.164479112 (error 0.73 percent).
Serial correlation coefficient is -0.000542 (totally uncorrelated = 0.0)
```

(This test data was collected on the bench under static conditions and is not representative but even so shows reasonable random characteristics)

16.4 Appendix D – Excerpts from RFC2104 - HMAC

We (3M) are frequently asked for an explanation of the authentication process used to mark Evidential Records. Full details (including references) may be found by searching the web for RFC2104. This extract describing HMAC is included to provide a local explanation of the authentication scheme in use and mitigate any concerns about the effectiveness of such a scheme.

Some authorities have requested the use of the RMAC authentication scheme. This is currently not recommended within the security industry. If required 3M can provide a description of the problems associated with RMAC and the advice followed in choosing the HMAC-SHA1 algorithm.

Network Working Group H. Krawczyk
HMAC: Keyed-Hashing for Message Authentication

Abstract

This document describes HMAC, a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, e.g., MD5, SHA-1, in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

1. Introduction

Providing a way to check the integrity of information transmitted over or stored in an unreliable medium is a prime necessity in the world of open computing and communications. Mechanisms that provide such integrity check based on a secret key are usually called "message authentication codes" (MAC). Typically, message authentication codes are used between two parties that share a secret key in order to validate information transmitted between these parties. In this document we present such a MAC mechanism based on cryptographic hash functions. This mechanism, called HMAC, is based on work by the authors [BCK1] where the construction is presented and cryptographically analyzed. We refer to that work for the details on the rationale and security analysis of HMAC, and its comparison to other keyed-hash methods.

HMAC can be used in combination with any iterated cryptographic hash function. MD5 and SHA-1 are examples of such hash functions. HMAC also uses a secret key for calculation and verification of the message authentication values. The main goals behind this construction are

- To use, without modifications, available hash functions. In particular, hash functions that perform well in software, and for which code is freely and widely available.
- To preserve the original performance of the hash function without incurring a significant degradation.
- To use and handle keys in a simple way.
- To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the underlying hash function.
- To allow for easy replacability of the underlying hash function in case that faster or more secure hash functions are found or required.

This document specifies HMAC using a generic cryptographic hash function (denoted by H). Specific instantiations of HMAC need to define a particular hash function. Current candidates for such hash functions include SHA-1 [SHA], MD5 [MD5], RIPEMD-128/160 [RIPEMD]. These different realizations of HMAC will be denoted by HMAC-SHA1, HMAC-MD5, HMAC-RIPEMD, etc.

Note: To the date of writing of this document MD5 and SHA-1 are the most widely used cryptographic hash functions. MD5 has been recently shown to be vulnerable to collision search attacks [Dobb]. This attack and other currently known weaknesses of MD5 do not compromise the use of MD5 within HMAC as specified in this document (see [Dobb]); however, SHA-1 appears to be a cryptographically stronger function. To this date, MD5 can be considered for use in HMAC for applications where the superior performance of MD5 is critical. In any case, implementers and users need to be aware of possible cryptanalytic developments regarding any of these cryptographic hash functions, and the eventual need to replace the underlying hash function. (See section 6 for more information on the security of HMAC.)

2. Definition of HMAC

The definition of HMAC requires a cryptographic hash function, which we denote by H, and a secret key K. We assume H to be a cryptographic hash function where data is hashed by iterating a basic compression function on blocks of data. We denote by B the byte-length of such

blocks (B=64 for all the above mentioned examples of hash functions), and by L the byte-length of hash outputs (L=16 for MD5, L=20 for SHA-1). The authentication key K can be of any length up to B, the block length of the hash function. Applications that use keys longer

than B bytes will first hash the key using H and then use the resultant L byte string as the actual key to HMAC. In any case the minimal recommended length for K is L bytes (as the hash output

length). See section 3 for more information on keys. We define two fixed and different strings ipad and opad as follows (the 'i' and 'o' are mnemonics for inner and outer):

```
    ipad = the byte 0x36 repeated B times
   opad = the byte 0x5C repeated B times.
```

To compute HMAC over the data `text' we perform

```
    H(K XOR opad, H(K XOR ipad, text))
```

Namely,

```
(1) append zeros to the end of K to create a B byte string
    (e.g., if K is of length 20 bytes and B=64, then K will be
     appended with 44 zero bytes 0x00)
(2) XOR (bitwise exclusive-OR) the B byte string computed in step
    (1) with ipad
(3) append the stream of data 'text' to the B byte string resulting
    from step (2)
(4) apply H to the stream generated in step (3)
(5) XOR (bitwise exclusive-OR) the B byte string computed in
    step (1) with opad
(6) append the H result from step (4) to the B byte string
    resulting from step (5)
(7) apply H to the stream generated in step (6) and output
    the result
```

For illustration purposes, sample code based on MD5 is provided as an appendix.

3. Keys

The key for HMAC can be of any length (keys longer than B bytes are first hashed using H). However, less than L bytes is strongly discouraged as it would decrease the security strength of the function. Keys longer than L bytes are acceptable but the extra length would not significantly increase the function strength. (A longer key may be advisable if the randomness of the key is considered weak.)

Keys need to be chosen at random (or using a cryptographically strong pseudo-random generator seeded with a random seed), and periodically refreshed. (Current attacks do not indicate a specific recommended frequency for key changes as these attacks are practically infeasible. However, periodic key refreshment is a fundamental security practice that helps against potential weaknesses of the function and keys, and limits the damage of an exposed key.)

4. Implementation Note

HMAC is defined in such a way that the underlying hash function H can be used with no modification to its code. In particular, it uses the function H with the pre-defined initial value IV (a fixed value specified by each iterative hash function to initialize its compression function). However, if desired, a performance improvement can be achieved at the cost of (possibly) modifying the code of H to support variable IVs.

The idea is that the intermediate results of the compression function on the B-byte blocks (K XOR ipad) and (K XOR opad) can be precomputed only once at the time of generation of the key K, or before its first use. These intermediate results are stored and then used to initialize the IV of H each time that a message needs to be authenticated. This method saves, for each authenticated message, the application of the compression function of H on two B-byte blocks (i.e., on (K XOR ipad) and (K XOR opad)). Such a savings may be significant when authenticating short streams of data. We stress that the stored intermediate values need to be treated and protected the same as secret keys.

Choosing to implement HMAC in the above way is a decision of the local implementation and has no effect on inter-operability.

5. Truncated output

A well-known practice with message authentication codes is to truncate the output of the MAC and output only part of the bits (e.g., [MM, ANSI]). Preneel and van Oorschot [PV] show some analytical advantages of truncating the output of hash-based MAC functions. The results in this area are not absolute as for the overall security advantages of truncation. It has advantages (less information on the hash result available to an attacker) and disadvantages (less bits to predict for the attacker). Applications of HMAC can choose to truncate the output of HMAC by outputting the t leftmost bits of the HMAC computation for some parameter t (namely, the computation is carried in the normal way as defined in section 2 above but the end result is truncated to t bits). We recommend that the output length t be not less than half the length of the hash output (to match the birthday attack bound) and not less than 80 bits (a suitable lower bound on the number of bits that need to be predicted by an attacker). We propose denoting a realization of HMAC that uses a hash function H with t bits of output as HMAC-H-t. For example, HMAC-SHA1-80 denotes HMAC computed using the SHA-1 function and with the output truncated to 80 bits. (If the parameter t is not specified, e.g. HMAC-MD5, then it is assumed that all the bits of the hash are output.)

6. Security

The security of the message authentication mechanism presented here depends on cryptographic properties of the hash function H: the resistance to collision finding (limited to the case where the initial value is secret and random, and where the output of the function is not explicitly available to the attacker), and the message authentication property of the compression function of H when applied to single blocks (in HMAC these blocks are partially unknown to an attacker as they contain the result of the inner H computation and, in particular, cannot be fully chosen by the attacker).

These properties, and actually stronger ones, are commonly assumed for hash functions of the kind used with HMAC. In particular, a hash function for which the above properties do not hold would become unsuitable for most (probably, all) cryptographic applications, including alternative message authentication schemes based on such functions. (For a complete analysis and rationale of the HMAC function the reader is referred to [BCK1].)

Given the limited confidence gained so far as for the cryptographic strength of candidate hash functions, it is important to observe the following two properties of the HMAC construction and its secure use for message authentication:

1. The construction is independent of the details of the particular hash function H in use and then the latter can be replaced by any other secure (iterative) cryptographic hash function.

2. Message authentication, as opposed to encryption, has a "transient" effect. A published breaking of a message authentication scheme would lead to the replacement of that scheme, but would have no adversarial effect on information authenticated in the past.

This is in sharp contrast with encryption, where information encrypted today may suffer from exposure in the future if, and when, the encryption algorithm is broken.

The strongest attack known against HMAC is based on the frequency of collisions for the hash function H ("birthday attack") [PV,BCK2], and is totally impractical for minimally reasonable hash functions.

As an example, if we consider a hash function like MD5 where the output length equals L=16 bytes (128 bits) the attacker needs to acquire the correct message authentication tags computed (with the _same_ secret key K!) on about 2**64 known plaintexts. This would require the processing of at least 2**64 blocks under H, an impossible task in any realistic scenario (for a block length of 64 bytes this would take 250,000 years in a continuous 1Gbps link, and without changing the secret key K during all this time). This attack could become realistic only if serious flaws in the collision behaviour of the function H are discovered (e.g. collisions found after 2**30 messages). Such a discovery would determine the immediate replacement of the function H (the effects of such failure would be far more severe for the traditional uses of H in the context of digital signatures, public key certificates, etc.).

Note: this attack needs to be strongly contrasted with regular collision attacks on cryptographic hash functions where no secret key is involved and where 2**64 off-line parallelizable (!) operations suffice to find collisions. The latter attack is approaching feasibility [VW] while the birthday attack on HMAC is totally impractical. (In the above examples, if one uses a hash function with, say, 160 bit of output then 2**64 should be replaced by 2**80.)

A correct implementation of the above construction, the choice of random (or cryptographically pseudorandom) keys, a secure key exchange mechanism, frequent key refreshments, and good secrecy protection of keys are all essential ingredients for the security of the integrity verification mechanism provided by HMAC.

16.5 Appendix E - Data structures for session protocol

(Not applicable to basic VES systems)

The protocol comprises two phases:

- Challenge – Response
- Session control

The challenge-response sequence *may* be issued independent of the session control protocol as a means of confirming and authenticating that the correct camera is online.
The challenge response protocol *must* be used as a prefix to the session control protocol.
(The data structures below are for guidance only. Please use supplied header files as definative document.)

```
ves_session.h
/***************************************************************************
 *        Copyright (c)       2005    Graham  Wood,   Pips Technology
 *
 *        The information enclosed herein is proprietary and is not
 *        to be reproduced in whole or in part without the express
 *        written permission of the Authors.
 ***************************************************************************
 */

// SESSION MANAGEMENT
// ==================

#define   USE_DH_EXCHANGE            0
#define   SHARED_KEY_LENGTH   256    // bits
#define   MAX_SECRET_LENGTH   64     // characters

static void    ves_derive_shared_key(char * site_id, dword serial, char * secret, char *
shared_key);

// challenge response protocol
// ---------------------------

// the shared secret may be distributed as a passphrase
// it is converted to an internal 256 bit number
// a seed is included to ensure reasonable bit depth even when the passphrase is short
// The result is stored as a hex numeric string representation of a 256 bit number

#define   SECRET_SEED         "the fountains mingle with the river"

// Instation system sends a packet of the form:
typedef p372_packed struct
{
         dwordcode;          // indicates challenge packet
         dwordlength;
         dwordseq;           // increments on each challenge
         byte nonce[16];     // random number
} gcc_packed VES_SESSION_CHALLENGE;

// The outstation replies with
typedef p372_packed struct
{
         dword code;         // indicates response
         dword length;
         dword seq;          // copied from challenge
         byte hash[20];              // response data
} gcc_packed VES_SESSION_RESPONSE;

// where the hash is created by performing an SHA1 hash of:
typedef p372_packed struct
{
         dword seq;          // copied from challenge
         byte site_id[32];   // null padded, null terminated string site identifier
         byte hash_secret[20];       // SHA1 hash of shared secret
         byte nonce[16];             // copied from challenge
} gcc_packed VES_SESSION_CHAP_DATA;
```

```
// The instation replies with

typedef p372_packed struct
{
         dwordcode;          // indicates success or fail
         dwordlength;
         dwordseq;           // original challenge
} gcc_packed VES_SESSION_CHAP_RESULT;

// If the response indicates fail then both
// the instation and outstation will close the connection
// if the response indicates OK_DONE, then no more work will take
// place and the connection may be closed
// If the response indicates OK_CONT then this challenge/response is a
// precursor to further interaction   on the same socket connection



// -----------------------------------------------------------------------

// key exchange - session control protocol
// --------------------------------------
// key data block may be AES encrypted with shared key
// shared key is either
//       transferred via dh protocol
//  or
//  derived from shared secret

// NB: data block must be multiple of 32 bytes
typedef p372_packed struct ves_keys
{
         byte nonce[16];
         byte magic[4];
         dwordsession;
         dwordstart_utc;
         dwordsession_length;
         char key1[64];
         char key2[64];
         char etz;           // added gaw 03 Oct ephemeral time zone - this session only
         char dst;           // boolean - set for daylight saving
         byte pad[10];
         byte digest[20];
} gcc_packed VES_KEYS;//ntoh store in network byte order (ves_start_session)

// extended key data structure to transfer/store keys
// for encryption of summary records and diagnostic/exception records

typedef p372_packed struct ves_keys_b
{
         byte nonce[16];
         byte magic[4];
         dwordsession;
         dwordstart_utc;
         dwordsession_length;
         char key1[64];
         char key2[64];
         char key3[64];
         char key4[64];
         char key5[64];
         char etz;           // added gaw 03 Oct ephemeral time zone - this session only
         char dst;           // boolean - set for daylight saving
         byte pad[10];
         byte digest[20];
} gcc_packed VES_KEYS_B;//ntoh store in network byte order (ves_start_session)


// to terminate a session the instation sends

typedef p372_packed struct
{
         dwordcode;
         dwordlength;
         dwordsession;
} gcc_packed VES_SESSION_TERMINATE;//ntoh

// to start a new session the instation send
```

```
typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          VES_KEYS     kb;
} gcc_packed VES_SESSION_NEW;//ntoh

typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          VES_KEYS_B   kb;
} gcc_packed VES_SESSION_NEW_B;//ntoh

// the outstation responds to either of the above with either

#define   VES_SESSION_MAGIC_FAIL             1
#define   VES_SESSION_SIG_FAIL        2
#define   VES_SESSION_CRC_FAIL        3
#define   VES_SESSION_INTERNAL_ERROR  4
#define   VES_SESSION_UNSYNCHRONISED  5
#define   VES_SESSION_TAMPER_SET             6
#define   VES_SESSION_EQUIPMENT_FAIL  7
// other reasons may be added as the system evolves)

// reject session request
typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          dwordsession;
          dwordreason;
} gcc_packed VES_SESSION_REJECT;//hton

// accept session request
typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          dwordsession;
          dwordutc;
          word  conf_crc;             // crc of system configuration
          word  app_crc;              // crc of system application
          dwordstatus_ext;            // extended status word
          dwordstatus;                // status word - system status at session start
} gcc_packed VES_SESSION_ACCEPT;//hton. set ubknown fields to 0

typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          dwordspare;
} gcc_packed VES_SESSION_SYNC_REQ;//ntoh

// time of daya at which routine sync check must take place
typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          dwordsecond;        // second after midnight to perform time check
} gcc_packed VES_SESSION_SYNC_TIME;//ntoh

// clock sync result
typedef p372_packed struct
{
          dwordcode;
          dwordlength;
          dwordsession;
          dwordutc;                   // utc from gps clock
          int   diff;                 // clock difference
          dwordstatus_ext;            // extended status word
          dwordstatus;                // status word
          int   result;               // +ve on success
} gcc_packed VES_SESSION_SYNC_RESULT;//hton

typedef p372_packed struct
```

```
{
          dwordcode;
          dwordlength;
} gcc_packed VES_SESSION_CODE;//hton/ntoh

// for efficiency
// data transfers are all of the same size

typedef   union
{
          VES_SESSION_CODE    ts_code;       // not real data - used to pull code field

          VES_SESSION_CHALLENGE       ts_chal;
          VES_SESSION_RESPONSEts_resp;
          VES_SESSION_CHAP_RESULT     ts_result;
          VES_SESSION_TERMINATE       ts_terminate;
          VES_SESSION_NEW             ts_new;
          VES_SESSION_NEW_B   ts_new_b;
          VES_SESSION_REJECT  ts_reject;
          VES_SESSION_ACCEPT  ts_accept;
          VES_SESSION_SYNC_REQ        ts_req;
          VES_SESSION_SYNC_RESULT     ts_sync;
          VES_SESSION_SYNC_TIME       ts_time;
}VES_SESSION_PACKET;


// session packet identification codes

// codes are
#define   VES_PKT_CHAP_CHAL           0xa5a50001
#define   VES_PKT_CHAP_RESP           0xa5a50002
#define   VES_PKT_CHAP_OK_DONE        0xa5a50003
#define   VES_PKT_CHAP_FAIL           0xa5a50004
#define   VES_PKT_CHAP_OK_CONT        0xa5a50005
#define   VES_PKT_DH_REQ                     0xa5a50006
#define   VES_PKT_DH_MYPK                    0xa5a50007
#define   VES_PKT_TERMINATE           0xa5a50008
#define   VES_PKT_NEW                 0xa5a50009
#define   VES_PKT_REJECT                     0xa5a5000a
#define   VES_PKT_ACCEPT                     0xa5a5000b
#define   VES_PKT_CLK_SYNC            0xa5a5000c
#define   VES_PKT_SYNC_RESULT         0xa5a5000d
#define   VES_PKT_SYNC_TIME           0xa5a5000e
#define   VES_PKT_NEW_B                      0xa5a5000f


// functions

static int             ves_process_challenge(int fd, VES_SESSION_PACKET * p);
static int             ves_process_ok_done(int fd, VES_SESSION_PACKET * p);
static int             ves_process_fail(int fd, VES_SESSION_PACKET * p);
static int             ves_process_ok_cont(int fd, VES_SESSION_PACKET * p);
static int             ves_process_dh_req(int fd, VES_SESSION_PACKET * p);
static int             ves_process_terminate(int fd, VES_SESSION_PACKET * p);
static int             ves_process_new(int fd, VES_SESSION_PACKET * p);
static int             ves_process_new_b(int fd, VES_SESSION_PACKET * p);
static int             ves_process_clock_sync(int fd, VES_SESSION_PACKET *p);
static int             ves_process_sync_time(int fd, VES_SESSION_PACKET *p);


#define   SESSION_STATE_IDLE  0
#define   SESSION_STATE_CHAP  1
#define   SESSION_STATE_CONT  2


// session tasks requests
#define   VES_SESSION_START   0
#define   VES_SESSION_STOP    1
#define   VES_SESSION_AUTH    3
#define   VES_SESSION_SYNC    4
#define   VES_SESSION_TIME_CHECK      5

 gcc_packed VKB;
```

16.5.1 Derivation of shared key

(Not applicable to basic VES systems)

The shared key must be unique to a specific camera. Therefore the shared key is derived from:

The shared secret - issued on deployment
The camera serial number - set by the factory
The configured site id - location specific

The shared secret may be delivered to the camera either as a binary sequence of 256 bits or as a passphrase of arbitrary length (< 64 characters).

When a passphrase is used, to ensure that a uniform shared secret is used internally, this passphrase is combined with a defined seed and hashed to generate an internal 256 bit representation of the secret. This also ensures that the passphrase cannot be reconstituted from the internal version of the secret. A code fragment to generate the internal form of the shared secret is:

```
#define   SECRET_SEED         "the fountains mingle with the river"

          bzero(secret, 64);
          SHA256Init(&tctx);
          SHA256Update(&tctx, SECRET_SEED, strlen(SECRET_SEED));
          SHA256Update(&tctx, passphrase, strlen(passphrase));
          SHA256Final(&tctx, d1);
          for(i=0,cp=secret; i<32; ++i, cp+=2)
               sprintf(cp, "%02x", d1[i]);
```

When creating the shared key from the shared secret, the serial number is represented as decimal eight digit zero padded and

the site id used is a concatenation of the short camera identifier and lane identifier, each string separated by underscore.

The shared key is then derived by:

a) generating an SHA256 hash of the internal shared secret
b) generating an SHA256 hash of the site_id and serial number
c) generating a further SHA256 hash of both these hash values.

SHA256 although processor intensive, is used here to maintain the 256 bit key depth. Multiple hash operations are used quite deliberately as this is recommended practise to make best use of the hash functions(S*chneier 2003*).

A code fragment to derive the shared key is:

```
// generate a hash of the secret
          SHA256Init(&tctx);
          SHA256Update(&tctx, secret, strlen(secret));
          SHA256Final(&tctx, d1);
// generate a hash of the site id
          SHA256Init(&tctx);
          SHA256Update(&tctx, site_id, strlen(site_id));
// add in system serial number
          sprintf(s,"%08d", serial);
          SHA256Update(&tctx, s, strlen(s));
          SHA256Final(&tctx, d2);

// generate a combined hash
```

```
          SHA256Init(&tctx);
          SHA256Update(&tctx, d1, 32);
          SHA256Update(&tctx, d2, 32);
          SHA256Final(&tctx, d1);
```

16.5.2 Summary and Evidential record data structures

```
/****************************************************************************
 *       Copyright (c)       2005    Graham  Wood,   Pips Technology
 *
 *       The information enclosed herein is proprietary and is not
 *       to be reproduced in whole or in part without the express
 *       written permission of the Authors.
 ****************************************************************************
 */

// Evidential & Summary record data structures

// system magic


// magic updated 24 Aug 05 to reflect ammended data structure
// magic updated 09 Dec 05 to reflect ammended data structure

// changes requested
// dword alignment
// app crc & config crc
// local time zone field
// add further 32 bits to status
// add 32 byte reserved area

#define   VES_MAGIC_BSR               "TL04"
#define   VES_MAGIC_SUMMARY   "TS04"
#define   VES_MAGIC_IMAGE             "TI04"
#define   VES_MAGIC_KEYS              "TK04"
#define   VES_MAGIC_RADAR             "TR01"
#define   VES_MAGIC_UIMAGE    "TU04"         // image transfer has unique key block
#define   VES_MAGIC_KB        "TB04"         // key block magic
#define   VES_MAGIC_DE        "TD04"         // diag/excep

#define   VES_MAGIC_KEYS_B    "TKB4"         // extended key transfer


#define   MAX_CTX             8


// lightweight summary record

#define   VES_MAGIC_TAGGED_SUMMARY    "TL01"

// a lightweght summary record has data transferred in the form
//        byte - (field type | field length)
//        array - variable length data field

// field lengths are
#define   VES_TAG_4           0x00
#define   VES_TAG_8           0x01
#define   VES_TAG_12          0x02
#define   VES_TAG_16          0x03
#define   VES_TAG_24          0x04
#define   VES_TAG_32          0x05
#define   VES_TAG_48          0x06
#define   VES_TAG_64          0x07

// tag identifiers are:
#define   VES_TAG_CAMERA              0x01
#define   VES_TAG_LANE        0x02
#define   VES_TAG_ORIENTATION 0x03
#define   VES_TAG_CRCS        0x04
#define   VES_TAG_SESSION             0x05
// etc

#define   VES_TAG_LENGTH_MASK 0x07
#define   VES_TAG_TYPE_MASK   0xf8
#define   VES_TAG_TYPE_SHIFT  3
```

```
// decoding process is to bzero the data structure, then bcopy each field into the
// data structure.

// simple data types will be larger, but arrays & complex entities may be abreviated

// a mechanism would be required (session protocol?) to transfer the required tag schema


// batched summary record - must be multiple of 16 bytes for inline encryption

typedef packed struct
{
          dwordsequence;             // event sequence number
          dwordfile_id;              // file id for er retrieval
          char plate[12];            // null terminated text plate reading
          byte confidence;           // confidence represented as a percentage
          byte class;                // vehicle classification
          byte status;               // status of system (as in heartbeat)
          dwordgps_secs;             // time stamp seconds
          word gps_ms;               // time stamp ms
          short diff_ms;             // signed difference between GPS and secondary times
          byte pad[1];
} VES_BSR;                           // exactly 32 bytes


// full record

typedef p372_packed struct ves_image_data
{
          char camera[24];           // camera identifier
          char lane[12];             // lane identifier
          char orientation[8];               // camera orientation
          word crcs[4];              // system crcs
          dwordsession;              // current session - zero if no session active
          dwordsequence;             // event sequence number
          dwordutc;                  // timestamp of primary event
          dwordutc_ms;
          int  local_tz;             // local time zone
          char country[8];           // null terminated text representation of country
          char plate[12];            // null terminated text plate reading

          int  flash;                // camera flash index
          int  gain;                 // camera gain index
          int  shutter;              // camera shutter index
          char zone[8];              // always returns "WEZ"
          dwordsr_length;            // dummy field required by ves - type & size tbd
          char mac[8];               // dummy field required by ves - type & size tbd
          char mobile_unit[12];      // dummy field required by ves - type & size tbd
          char media_source[12];     // dummy field required by ves - type & size tbd
          int  direction;            // direction of travel of target - to/from/notsure
          int  number_of_images;     // number of images transferred
          byte oem_data[VES_OEM_DATA_SIZE];  // 64 bytes

          int  confidence;           // confidence represented as a percentage
          int  conf_cat;             // confidence represented as a statistical category
          dwordclass;                // vehicle classification
          dwordfield_number;         // field number of image
          dwordstatus_ext;           // extended status bits
          dwordstatus;               // status of system (as in heartbeat)
          int  ctx_count;            // number of context images
          int  ctx_offsets[MAX_CTX]; // time offset mS of context images
          int  xx1;                  // x-coord of plate find 1
          int  yy1;                  // y-coord of plate find 1
          int  xx2;                  // x-coord of plate find 2
          int  yy2;                  // y-coord of plate find 2
          int  traj_interval;        // time interval mS between find 1 & find 2
          char site_description[100];// free text site description
// reserved 32 bytes
          dwordgps_secs;             // gps time stamp seconds
          dwordgps_ms;               // gps time stamp ms
          dwordgps_lat;              // GPS position as signed decimal degrees
          dwordgps_long;             //  x 10 ^6
          byte reserved[16];
}gcc_packed  VES_DATA;
```

```
typedef p372_packed struct ves_image_sizes
{
         int  patch_size;
         int  full_size;
         int  oview_size;
         int  ctx_sizes[MAX_CTX];
} gcc_packed VES_SIZES;


// image data goes here - variable length
// patch, full ir, oview, ctx1, ctx2 ...




// if bvom then the data block is repeated for the client camera
// ie another VES_DATA followed by another VES_SIZES
// followed by padding - ER is padded to multiple of 32 bytes (inc digest)
// first byte of padding will have size of padding - minimum pad size is one byte
// then SHA1 digest (20 bytes)

// all data is transferred in NETWORK BYTE ORDER



// OEM data structures

// these are mapped onto the oem data array so must not exceed 64 bytes


typedef p372_packed struct ves_speed_read
{
         dwordmagic;

         dwordutc;
         word utc_ms;
         shortspeed;

// data used to confirm radar configuration
         byte direction;
         byte units;
         word correction_divisor;

} gcc_packed VSR;


// if a system must use a unique key pair for each ER transferred, then the key block
// is transferred between the header and the ER proper
// NB: there is no CRC is this block. This is deliberate. A CRC would give an attacker
// clear indication that a key pair had been successfully retreived.

typedef p372_packed struct ves_key_block
{
         byte nonce[16];
         byte magic[4];
         byte reserved[12];
         byte key_1[32];
         byte key_2[32];
} gcc_packed VKB;

// all data is transferred in NETWORK BYTE ORDER
```

16.5.3 Transfer of shared secret

There are two implemented options for transfer of the shared secret. The secret may be transferred either as a passphrase entered manually or an encrypted string passed via a machine interface.

These schemes are mutually exclusive. A system will be built to use one system or the other. Whichever system is used, the data *must* be entered via the serial port. Any attempt to enter the shared secret via another route (e.g. telnet or web page) will be rejected.

The passphrase once entered, is passed through a one way hash function and subsequently cannot be recovered from the camera.

16.5.4 Manual entry

The CLI provides a command set `ves secret prime`. The operator enters the shared secret. Entered characters are not echoed. The operator is requested to renter the secret. If both entries match the secret is accepted.

```
>> ves secret prime ...
```

To enter an interactive process to set the shared secret the commands are:

```
>> ves secret prime ...
>> ves secret trial ...
>> ves secret check
```

Where `prime & trial` set the appropriate secrets.

`check` will read back both secrets and confirm that they are intact.
(They are not printed!)
Under normal circumstances the trial secret will never be accessed. However the application software may be built to only use the trial secret when initial deployment cannot meet the deliverable security constraints.

An example usage is:

```
>>ves secret
Available Commands are:
        prime$
        trial$
        check$
>>
>>ves secret prime
Set site secret
new secret: ******
     confirm: ******

* CMD:OK
>>
>>ves secret trial
Set site secret
new secret: *****
     confirm: *****

* CMD:OK
>>
>>ves secret check
VES prime secret is OK
VES trial secret is OK
* CMD:OK
>>
```

16.5.5 Machine entry

The system expects a strictly formatted string of the form:

```
          ves secret prime data
or
          ves secret trial data
```

Where:

Field	*Description*
Data	The data field – ascii text representation of hex digits – fixed length - 64characters shared secret, 4 characters CRC secret.

The 64 character shared secret component is encrypted with AES256. The encryption key is formed by performing a hash of the camera site id and camera serial number. The data block includes an appended CRC. The camera will decrypt the data block then check the CRC. The purpose of this exercise is not primarily to hide the shared secret, but rather to ensure that the secret is received by the correct camera and is received intact.

When the shared secret is entered in this form the shared secret is assumed to be already converted to an internal form ie a 256-bit number.

Example code fragment to process shared secret input

```
          int  l;
          char buffer[32], keystr[66],key[32];
          word crc,newcrc;
          char *cp;
          int  i;
          SHA1_CTX      tctx;
// fixed format input
// the secret is encrypted to ensure that the correct key for this camera is
// supplied - this is not a primary security measure.
               if((l = strlen(argv[0])) != 68)
               {
                       printf("incorrect arg length (%d)\n", l);
                       return CMD_FAIL;
               }
               bzero(buffer, 32);
// convert ascii to binary
               load_key(buffer, argv[0], 256);
               load_key((char *) &crc, argv[0]+64,  16);
// create a decrypt key based on serial number and site_id
               bzero(keystr, 64);
```

```
                sprintf(keystr,"%08x%s",serial.serial_number,ves_get_site_id());
                SHA1Init(&tctx);
                SHA1Update(&tctx, keystr, strlen(keystr));
                bzero(keystr, 64);
                SHA1Final(key, &tctx);
// ascii representation of key
// NB: key is 256 bits, but last 80 bits will be zero
                for(i=0,cp=keystr; i<32; ++i,cp+=2)
                        sprintf(cp, "%02x", key[i]);
// decrypt the secret
                ves_crypt((char *) buffer, 32, keystr, VES_DECRYPT);
// compute crc
                newcrc = 0;
                compute_crc_16((byte* ) buffer, 32, (word *) &newcrc);
                if(crc != newcrc)
                {
                        printf("Corrupt or incorrect secret (crc fail)\n");
                        return CMD_FAIL;
                }
// generate the internal ascii representation
                for(i=0,cp=secret; i<32; ++i, cp+=2)
                        sprintf(cp, "%02x", buffer[i]);
                bzero(buffer, 32);
```

16.6 Appendix F – Example Software update procedures

The P372 program images are stored in boot flash memory in a simple file system. A software update procedure can be as simple as copying a new file into the boot flash system then executing a restart.

However when systems are remote, there are many such systems, multiple files must be replaced reliably and configuration options changed then an automated procedure to initiate and manage updates is extremely desirable.

Such a set pf procedures exists for the P372 camera system. The core of this system is the "PIP" file.

16.6.1 PIP Files

PIP Files are files created for installing into the P372 system. A PIP file contains one or more components of the software to be installed on a P372, and is packaged in such a way that the system knows how to install the software itself.

At minimum a PIP file can contain a single file such as a flex file or script file - in which case it is just a wrapper that allows the system to install that file with no further instructions. At the other extreme a PIP file can contain all of the software and configuration details to take a system and configure it exactly as necessary from scratch.

This appendix is aimed at users who wish to create or manipulate PIP files and already have knowledge of 357/372 files.

16.6.2 Tools

There are two tools provided to work with PIP files:

1)

These tools are written in perl. Perl is available for most platforms. These tools have been tested on Windows XP with Activestate's perl (available as a free download from http://www.activestate.com). Linux and Mac OS X both usually come with a suitable perl.

To install a pip file it must first be uploaded onto the system. As PIP files are usually large so drive sd0: is the best place. A manual install procedure would be:

```
>> cd sd0:
>> ftp get file.pip
 >> install file.pip
```

A PIP file may also be installed transparently from the web page.

16.7 Appendix G - Readability of License Plates

16.7.1 Terms Specific to the 3M System

A character is said to be *obstructed* by an object, if the silhouette of the object in question overlaps any portion of the character. Obstructions are caused by objects lying in the path between the camera and the license plate.

Also a character is said to be *obscured* by an object, if the object in question casts a shadow which overlaps any portion of the character. A *shadow* is defined as any portion of the character region in which the average light grey level is 6% darker (lower in value) than the average light grey level for the rest of the character region. Obscurations are caused by objects lying in the path between the light source which may be either the illuminator or the sun (depending upon prevailing conditions) and the license plate.

A character is said to be *broken* if either its silhouette is composed of two or more disconnected components, or the connectivity of holes in the character is changed in any way (e.g. two holes are connected, or a hole is connected to the background).

16.7.2 What is Not Readable?

Obviously, if the vehicle's license plate is read correctly, then the license plate is readable. The following tests are applied when a license plate is not read correctly to determine if the license plate was readable:

- License plates not mounted on the vehicle in the manufacturers designated location or not firmly secured to the vehicle in that location are not considered readable.

- License plates that contain one or more broken characters (e.g. as the result of being damaged) are not considered readable.

- License plates with contrast levels below 15% of grey scale are not considered readable. The grey scale for 3M License Plate Reader (ALPR) corresponds to the interval [0, 255], thus, the contrast must be greater than 30 grey levels for the license plate to be considered readable. Deposition of salt on license plates during winter months can result in reduced contrast levels on an otherwise good plate.

- License plates on which one or more characters are obstructed or obscured are not considered readable.

- License plates on which the character region is either obstructed or obscured by such things as road soil, snow, corrosion, scratches, tow hitches, bumpers, items hanging from or mounted on the vehicle, or any other items, are not considered readable. A thick layer of road soil can cause an obstruction by completely covering the surface of the plate on a particular location. A thinner layer of road soil, on the other hand, may not qualify as an obstruction, but could reduce the contrast of the license plate below the 15% threshold. Corrosion, snow, mud, etc. can cause portions of the characters be obstructed. Tow hitches can create obstructions, Obscurations or a combination of both.

- License plates, which are otherwise considered illegal, whether illegally mounted, illegally manufactured or illegally configured are not considered readable.

- Vehicles being towed are not considered readable.

16.7.3 What is an Incorrect Read?

A license plate which meets all other readability criteria, shall be considered incorrectly read whenever one of the following conditions applies:

- the ALPR incorrectly identifies any of the characters in the license plate (the wrong character is reported);

- the ALPR fails to recognise the existence of a character (no symbol is reported at all), or an extra character is added to the license plate license where no corresponding character exists.

Characters that are not identified correctly will not be considered incorrectly read unless the license plate meets all other readability criteria.

16.7.4 Required image size & orientation

The ALPR will require an image of adequate size in order to resolve the details within each character. This may be considered the "optimum" size. As the license plate image becomes smaller than optimum, read performance will deteriorate. As the license plate image becomes larger than optimum processing time will increase. In application involving high speed vehicles or high volumes of traffic where the system is time constrained this will also lead to deteriorating performance. The OCR system is optimised for the smallest reasonable size of plate since this will give the largest possible field of view for the camera.

16.7.5 Optimum size

The minimum plate size that the captured image can be is 90 pixels horizontally and 12 pixels vertically. However at this size plate read reliability will be poor as features are easily disrupted. The optimum image size will be in the region of 150 pixels horizontally or 18 pixels vertically.

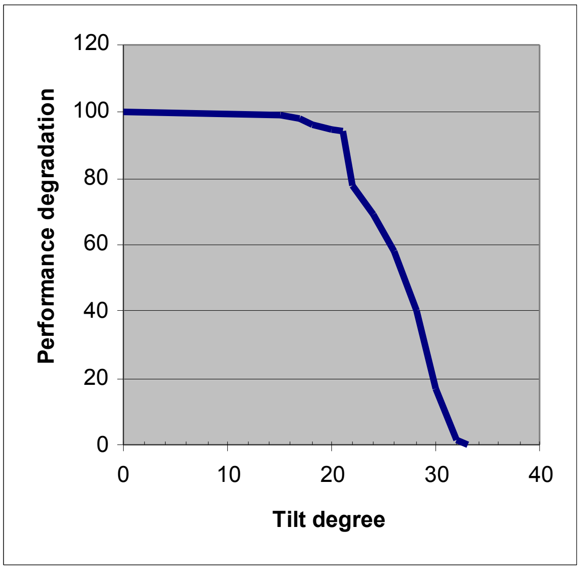
The optimum horizontal plate size as a percentage will vary depending on the sensor used in the camera. 3M uses various sensor sizes depending on camera model.
For the 372 family cameras the image width is 720 pixels. On this camera the optimum plate image falls between 20%-30% of the horizontal image width. 3M require that plate images are at least 20% of image width to provide specified read rates.

For 382 family cameras (as used in the Speedspike system) the image width is 1392 pixels. On this camera the optimum plate image falls between 10%-15% of the horizontal image width. 3M require that plate images are at least 10% of image width to provide specified read rates.

Optimum orientation

Ideally a plate will be aligned in the field of view such that the edges of the plate image are parallel with the edges of the imaging device. However this is not always practicable.

A plate is tilted when it is rotated within its own plane. (i.e. about an axis parallel with the normal forward direction of travel of the vehicle). A plate is skewed when it is rotated about any other axis.



As shown above, the plate may be tilted ±10 degrees without any significant deterioration in read rate. As tilt is increased beyond 10 degrees the read rate will start to suffer. If the plate is tilted beyond ±30 degrees the ALPR will cease to operate in a meaningful fashion.

The plate may be skewed in the field of view without any significant deterioration in read rate provided that:

- the distortion in character size is such that the maximum and minimum character size still meet the size constraints described above
- the plate still returns an image meeting the contrast requirements described above.
- The retro-reflectivity factor is still adequate. This can vary from country to country.

17 VES ADDENDUM - Radar Interface

17.1 Overview

This is a brief overview of the VES radar interface. This addendum is part of the VES Interface Specification.

3M are able to supply cameras with a doppler radar assembly. This radar may be used in conjunction with the VES software to record the speeds of individual vehicles passing through the field of view. Evidential records may be generated for all vehicles exceeding a predetermined speed.

The radar module may be fitted to any camera which has an auxiliary serial port uncommitted. This may exclude any camera equipped with a GPRS modem.

If supporting context images are required in an enforcement application 3M recommend use of a camera with the full VES interface and hardware jpeg capability.

17.2 Specification

description	***value***
OPERATING FREQUENCY:	34.7 GHz (Ka-band)
STABILITY:	±100 MHz
POWER REQUIREMENTS:	Voltage: 9 - 16 VDC Current (at 12 VDC nominal) with transmitter on: 370 mA
ENVIRONMENTAL:	Operating: -30°C to +70°C, 90% relative humidity Non-operating: -40°C to +85°C
MECHANICAL:	Weight – 1.15 lb. (0.52 kg) Diameter – 2.6 in. (6.7 cm) Length – 4.7 in. (11.8 cm) Case Material – Aluminium die cast
ACCURACY:	+1, -2 MPH +1, -2 KPH
AUTO SELF-TEST:	Performed every 10 minutes while transmitting
SPEED RANGE:	Stationary low speed threshold: 12 MPH to 200 MPH (19 to 321 KPH)
ANTENNA:	Conical horn
POLARIZATION:	Circular
3DB BEAMWIDTH:	12° ±1°
POWER OUTPUT:	10 mW mininum 15 mW nominal 25 mW maximum
POWER DENSITY:	1 mW/cm2 maximum at 5 cm from lens

17.3 Configuration

The main VES configuration section has two applicable parameters:

Command	Parameter	Value	Decsription	notes
ves set / show	radar_enable	0,1	When set enables the radar interface. Changing this parameter requires a reset to take effect.	
	radar_threshold	nnn	When vehicle speed is above this threshold then an ER may be generated. Set the threshold to zero to send ERs on all events. NB on a full VES system with sessions enabled, an ER will only be generated when the camera is in session. (Summary records will always contain speed data when it is available and will always be forwarded)	

When the radar system has been enabled and the system reset, then a new parameter table automatically becomes available to manage the radar sub assembly:

Command	Parameter	Value	Description	notes
radar set / show	units	n (0-2)	This parameter selects units used for configuration and reporting 0-imperial 1-uk 2-metric	1
	sensitivity	n (0-16)	Set the radar sensitivity range 0-16	1,2
	direction		Specify whether vehicles speed is measured for vehicles approaching or receding from the camera/radar 0 - receding 1 - approaching	1
	acquisition	0xaaaa	Should not normally need to be changed. Sets the ratio of good reads required to establish radar lock. Format is x of y, x in high byte, y in low byte where target acquistions requires >=x of the last y buffers to have good signal to noise ratio. For example the default 0x080a indicates that 8 good reads in 10 are required.	1
	loss	0xaaaa	Should not normally need to be changed. Sets the ratio of bad reads	

			required to lose radar lock. Format is x of y, x in high byte, y in low byte where target loss requires >x of the last y buffers to have bad signal to noise ratio. For example the default 0x020a indicates that 2 failed reads in 10 are required.	

	`agc_min`	`n`
		`(0-6)`
			Should not normally need adjustment
			Sets minimum AGC level default 0. Must be less than maximum agc gain.	

	`agc_max`	`n`
		`(1-7)`
			Should not normally need adjustment.
			Sets maximum AGC gain. Must exceed minimum agc gain. default 7	

	`height`		height of camera above ground	`1,4`
	`range`		range from camera to expected location of plate Set range to zero to disable correction.	`1,4`
	`offset`		offset between camera position and centre of vehicle lane	`1,4`
	`window`	`nnn`	The message from the radar must match the plate read timestamp within this window	`1,3`
`radar display`			Display the internal configuration state off the radar	
`radar calibrate`	`on/off`		switch radar in/out of calibrate mode. When in calibrate mode, no speed data will be store or returned to the ANPR capture system. The radar subassembly always starts in calibrate mode - this mode is then disabled by the camera software. When switched into calibrate mode the system will automatically switch out of calibrate after a period of 7 minutes.	

notes:
1) These commands/parameters will only become available when the ves parameter `radar_enable` has been set and the system has been reset.

2) radar sensitivity may need to be adjusted from default based on camera range and expected vehicle size.

3) The radar generates a new speed reading approximately 20 times per second. The camera records and timestamps each of these readings, then locates and matches the reading closest to the image used for ANPR and reported as the full IR image or patch image. If a speed reading cannot be found, then no speed data will be generated. If the radar was unable to lock onto the target vehicle then the speed reading will be present but will indicate 0.

4) These distances should be entered in the units system specified for speed. However what matters is that all units are to the same scale ie all in feet, all in metres or even all in millimetres. Fractional units are not accepted.

17.4 System hardware and interconnection

The radar module will be supplied preconfigured by 3M to operate with the camera interface. Once connected to the camera, further site specific radar configuration may be required.

In particular, the camera site information (height, offset and range) must be set correctly to apply the appropriate correction.

17.5 Usage guidelines

If the system is to be used for enforcement purposes, then, for the avoidance of doubt, if an evidential record image set shows any vehicle other than the target, then this event should be downgraded and possibly not used for enforcement.

Ideally the camera and radar should be sited such that only one lane is monitored, and that lane should be straight and clear within the monitored area.

Many jurisdictions will require an alternate indication of vehicle speed to resolve doubt in contested situations. This may be taken from the supporting context images - normally captured approx 240ms before and after the ANPR event though this may be configured on site. These images are only available on SPIKE+ system with hardware jpeg and FULL VES capability.

See appendix to this addendum for further guidance notes.

17.6 Mounting and alignment

(to be completed)

The radar module is mounted on a bracket under the camera. This bracket allows for small adjustment to the vertical radar angle. The radar should be adjusted so that its field of view includes the field of view of the camera but minimises long range views of the road. Generally, if in doubt, the axis of the radar should be parallel with the axis of the camera.

A cable is provided to link the radar module to the camera. This cable provides both power and signal connections. As the radar module couples to on the socket at the back of the camera this will necessarily restrict possible connections to the camera.

The radar module has a maximum supply rail of 16V (and draws 300mA). Care must be taken to ensure that the camera supply rail does not exceed 16V. The standard supply provided with a P372 supplies 18V nominal. On a long cable run (>25M) it is likely that the 16V limit will not be exceeded. However on short cable runs the 18V supply should be substituted with a 15V unit.

(drawing of mounting bracket)

(drawing of cable)

(Connections available on breakout box)

17.7 Data transfer

The radar data is transferred in the OEM field of all summary and evidential records.

The format of this data is:

```
typedef packed struct ves_speed_read
{
      dwordmagic;
      dwordutc;
      word  utc_ms;
      shortspeed;

      byte  direction;
      byte  units;
      word  correction_divisor;

} VSR;
```

Magic will be set to the unterminated four byte string: `TRnn` - initially `TR01`.

When the correction is applied to the speed, to maintain operation within an integer space, the camera multiplies the measured speed by 1000, then divides by the correction divisor. If correction is not enabled (range set to 0) then the divisor will be set to 1000.

All data is in network byte order.

If an SR, ER is expected to have a speed indication, but the magic field is not found then this is an indication that no speed value could be found which matched the vehicle detected. This will occur whilst the radar is being calibrated, configured or reconfigured. It will also occur if the radar is not connected or is malfunctioning.

The camera system will select and match the closest radar speed reading to the captured IR image. These readings cannot be entirely coincident. The radar will generate approximately 20 readings per second. The camera will capture images 50 time per second. The radar specific OEM data set contains the actual radar capture time. The SR and ER contain the actual capture time of the IR image. This allows the back office system to make a further check to ensure that image capture time and radar capture time relate to the same event.

If text overlays are enabled, then the monochrome images (patch and full IR) may show the vehicle speed. The colour overview will not carry vehicle speed.

Speed is displayed to the nearest whole number in the specified units.

Units may be specified as:

unit type	***description***
`imperial`	Speed is measured in mph. distances are measured in feet.
`uk`	Speed is measured in mph, distances are measured in metres.
`metric`	Speed is measured in kph, distances are measured in metres.

17.8 Approvals

The doppler radar module is certified by the Institute for Police Technology Management (Florida) as accurate within +1/-2 mph when tested to the standards required by the International Association of Chiefs of Police .

This system has not been approved for enforcement purposes in the UK and therefore may only used on public roads in the UK for speed advisement and survey purposes.

This radar may not be approved for use in US military reservations or within the boundaries of any airport. Contact 3M before using this radar in such locations.

17.9 Error minimisation & correction

17.9.1 Cosine correction

The radar assembly is built to read the speed of vehicles directly approaching or receding from the sensor. If the sensor is mounted above and / or to the side of the carriageway then the sensor will appear to under read.

The Doppler principle on which the radar depends, explains the frequency shift associated with energy waves reflected by or emanating from a moving body.

In the case of the P372 radar subassembly a Ka band radar signal is transmitted at a specific frequency by the sensor, reflects off of a target (or targets) and returns to the sensor . If either the sensor or the target are moving relative to one another, the signal will be shifted in frequency when it returns to the sensor. This shift in frequency allows measurement of the relative velocity between the sensor and target.

The fundamental Doppler frequency shift is given by:

```
    F_shift = (2 * V * F_radar * cos(q)) c
```

where:

variable	*description*
F_shift	Doppler shift Hz
c	Velocity of light
V	relative velocity
F_radar	Radar frequency (34.7 ± 0.1 GHz (Ka Band))
q	offset angle of sensor relative to direction of target motion

17.9.2 Correction for Offset Angle

Any offset angle between the centre of the radar beam and target direction of travel will introduce a factor of cosine (q) into the measured speed. This means that the output of the sensor must be corrected by dividing into it the cosine of the offset angle.

The correction required (known as the cosine correction) may be applied by the radar interface software.

```
true speed = indicated speed / cosine q
```

where q is the angle between the beam path and the vertical and/or horizontal displacement. By configuring the camera to include height, offset and range the camera can correct the cosine error.

The radar beam diverges about 6° from centre, resulting in a roughly conical-shaped beam. In the case of a target passing a fixed sensor, this geometry can introduce a further "cosine error" into the speed measurement. This happens because targets at one edge of the beam are at a different offset angle than in the centre of the beam. For small offset angles, the cosine change from one edge of the beam to the other is small and so the cosine error is minimal. For larger offset angles, the change is more significant. No correction can be applied for this error.

Because the value of cosine changes rapidly for offset angles above 45°, these angles are not recommended.

17.9.3 Signal Strength and Multiple Targets

The radar includes a signal processing algorithm that determines the strength of return signal from a target. If the signal is strong enough, the output is turned on and the sensor is said to be "locked". Because different targets reflect different amounts of the radar energy back to the sensor, the sensor will lock at different distances from the target depending on such factors as target size, material and orientation.

In general, large targets reflect more energy and the sensor will be able to distinguish them at a greater distance.

The sensor receives reflected energy from all possible targets within the radar beam. If any of the targets are moving, it will cause a Doppler shift, possibly causing a false measurement if it is not the desired target. For this reason, it is important to consider the beam geometry, particularly the divergence angle, and make sure that the sensor cannot "see" non-targets.

17.9.4 Calibration / Test

(This section is not yet implemented)

As can be seen from above, the radar may be checked or calibrated by generating a doppler signal via mechanical modulation. Often this is done using a tuning fork. There are also PC applications which may be used for lab tests but these are less appropriate for roadside checks.

The required tuning fork / test freq. may be calculated from:

```
F_fork = (2 *  347 * V) / 3
```

```
(V in M/sec)
```

test speed	***as m/s***	***freq***
30 mph	13.41	3102
50 mph	22.35	5170
40 kph	11.11	2570
80 kph	22.22	5140

To check calibration with either a tuning fork or a PC calibration package the system must be placed in calibration mode (command `radar calibrate`). This instructs the radar to accept both approaching and receding signals. It also ensures that evidential records containing erroneous reading will not be generated.

Data reported in calibration mode has no cosine correction applied.

Place either the PC speaker or the vibrating tuning fork directly and squarely in front of the radar dome at a distance of about 30 cm (1ft). The camera terminal will report the calibration speed.

17.10 APPENDIX - Further Guidance notes

The following guidance notes are extracted from:
Code of practice for operational use of enforcement equipment © ACPO 2002 Version
This document is an extract and does intend to represent the complete original. Interested parties may care to examine the original documents.

17.10.1 Radar Coverage

The radar will always display the speed of the strongest returned signal. When two cars of similar size are approaching the meter it **may** display the speed of the nearer vehicle. The signal received by the radar not only depends on the distance of the vehicle, but also its size, so that a large truck some distance from the meter may return a signal stronger than a small car closer to the meter. The radar meter has an aerial which forms the radar signal into a beam so most of the signal is sent out directly ahead of the meter, but some of the signal spreads out on either side of the main beam. The coverage of the radar depends on several factors, including:

* the radar power
* receiver sensitivity
* aerial characteristics
* the size of the target vehicle
* distance from the observer,
* its position in the aerial beam

17.10.2 Cosine Factor

Radar will only record the true speed if the radar is in the line of the vehicle’s path. The vehicle may be either approaching or receding. If the radar is positioned at an angle to the path of the vehicle, the apparent speed of the vehicle is reduced. The reduction in speed is proportional to the cosine of the angle. For an angle of 15 degrees a speed of 38mph will be recorded for a vehicle travelling at 40mph. The cosine factor is always in favour of the driver. Roadside radar devices have inbuilt electronic circuits to correct the cosine factor provided the unit is aligned correctly in accordance with manufacturer’s instructions.

17.10.3 Site Selection

Always choose a site that affords road users maximum safety with regard to any potential hazard. As described previously, the radar signal is reflected from a moving vehicle. The signal can also be reflected or scattered by stationary objects such as road signs, hoardings, stationary vehicles near to the radar, or pillar-boxes. As these objects are stationary, no Doppler effect should occur and no readings appear on the display. However, these objects can act as radio 'mirrors' and reflect signals from moving vehicles outside the area of the coverage diagram . For example, the signal could be reflected around a bend in the road and
measure the speed of a vehicle not visible to the radar operator. Because of the scattering effect, signals returned to the radar in this manner will be very weak and far less than the signal returned from a vehicle directly in the radar coverage area. While it is extremely unlikely a reflected signal will have any effect while a vehicle measurement is being made sensible precautions should be taken to select a site free from reflecting objects and with a clear, unobstructed view of the road. The ability of objects to reflect radar signals varies. A flat metal surface, such as a hoarding, will reflect more efficiently than a run of trees, which would
absorb and scatter the signal.

To act as a radar mirror, objects may only need to:
- be at the correct angle

- rotate at the correct speed
- occupy a portion of the field of view
- be reflecting a fairly strong signal.

Always select a site with a clear view of the oncoming traffic and which is free of any large objects such as: bus shelters, large road signs, fences/crash
barriers, stationary large vehicles.

To avoid multiple reflections the radar must not be situated under a bridge or arch and should not be targeted through bridges, railway arches or concrete lined cuttings.

17.10.4 Summary

The site must be tested and shown to be clear of any obvious source of interference and reflectors.
The equipment must have a clear view of the road.

17.10.5 Radio Interference

It is impossible to ensure complete immunity from radio interference.

If radio interference is present the speed display may still show a speed reading. The reading may be steady, or erratic depending upon the type of transmitter. Interference effects will only occur when the radar is close to the transmitter or the transmitter is very powerful. It is not possible to lay down strict criteria for safe operating distances from some transmitters. The strength of the interference depends on several factors, such as transmitter frequency, type of aerial and modulation system.

When selecting a site, **treat with extra caution** places with view of: high voltage overhead lines, transmitting masts or tower, airports or harbours, and any other place where high power radar transmitters may be expected to operate. Never point a radar at a civil or military aircraft, vessel or armoured vehicle. Many military aircraft, vehicles or vessels have target acquisition detectors, some of which initiate automatic counter measures. With the current popularity of Citizen Band radio transmitters, mobile telephones and satellite or radio communication systems, it may be difficult in a suburban street to tell the difference between transmitting aerials and domestic receiving aerials. It is important, therefore, to ensure no interference is present by carrying out 360º checks.

17.10.6 Multiple Vehicles

Radar speed meters are designed to measure the speed of one vehicle at a time. Should there be more than one vehicle present in the radar field of view, it is possible for the device to detect two different signals, and
alternately display different speeds in which case the check **must** be aborted. With more than one vehicle (especially when they are of a similar size) within the range of the radar, the meter may read the nearer vehicle, but not necessarily, since the reflected signal from a vehicle is very complex and fluctuates rapidly as the view of the vehicle changes slightly. **An operator must not make detections for prosecution when more than one vehicle is within the radar detection range.** It is quite possible for the signal from a large vehicle some distance behind a smaller vehicle to override the signal from the nearer vehicle.

The ideal enforcement situation is when only one vehicle is isolated in the radar field. Under no circumstances must other moving vehicles, travelling in any direction, be between the radar and the target vehicle. If for any reason the operators have any doubt as to the validity of the reading, the check must be abandoned.