**3M United Kingdom PLC**

3M

Кейн роуд Брекнел Бъркшир RG12 8HT

Великобритания

# Спецификация на софтуерен интерфейс на система за разпознаване на превозни средства (VES)

### СПЕЦИФИКАЦИЯ НА СОФТУЕРЕН ИНТЕРФЕЙС VES ИЗДАНИЕ 1V

Информацията, която се съдържа в този документ е конфиденциална и е собственост на 3М United Kingdom PLC. Съдържанието не може да се разкрива на трети страни без изричното писмено разрешение на 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

# Съдържание

[1 Преглед на документа 5](#_Toc390256951)

[1.1 Въведение 6](#_Toc390256952)

[2 Общо конфигуриране 8](#_Toc390256953)

[2.1 VES специфично конфигуриране 8](#_Toc390256954)

[2.2 Идентификация на местоположението 15](#_Toc390256955)

[2.3 Синхронизация на времето 16](#_Toc390256956)

[2.4 Валидизация на системната конфигурация 19](#_Toc390256957)

[3 Последователност на събитията 20](#_Toc390256958)

[3.1 Протокол на изпращане 20](#_Toc390256959)

[3.2 Протокол на изтегляне 20](#_Toc390256960)

[3.3 Локално запаметяващо устройство за доказателствени записи 21](#_Toc390256961)

[3.4 Допълнителни снимки 22](#_Toc390256962)

[3.5 Данни за траекторията 22](#_Toc390256963)

[Guidance on expected file sizes 24](#_Toc390256964)

[4 Streaming interface 25](#_Toc390256965)

[4.1 Image record 25](#_Toc390256966)

[4.2 Summary record 28](#_Toc390256967)

[4.3 Batched Summary records 28](#_Toc390256968)

[4.4 Acknowledgement 30](#_Toc390256969)

[5 NVT channel 31](#_Toc390256970)

[5.1 ASCII interface 31](#_Toc390256971)

[5.3 Binary interface 31](#_Toc390256972)

[6 Session protocol 35](#_Toc390256973)

[7 System Security & data Integrity 36](#_Toc390256974)

[7.1 Key exchange 36](#_Toc390256975)

[7.2 Outstation authentication 37](#_Toc390256976)

[7.3 Shared secret 37](#_Toc390256977)

[7.4 Outstation integrity 37](#_Toc390256978)

[7.5 System self check 38](#_Toc390256979)

[7.6 Port protection 39](#_Toc390256980)

[7.7 Comments on compliance with PSDB guidelines 41](#_Toc390256981)

[8 Fault Tolerance - Host detection & communications failure 43](#_Toc390256982)

[9 Message channel 45](#_Toc390256983)

[9.1 Encrypted Envelope 45](#_Toc390256984)

[9.2 Plain text 46](#_Toc390256985)

[10 Heart beat Channel 48](#_Toc390256986)

[10.1 Status report 48](#_Toc390256987)

[11 Trigger Channel 51](#_Toc390256988)

[12 Image streaming protocols 52](#_Toc390256989)

[12.1 Viewfinder 52](#_Toc390256990)

[12.2 Snapshot 52](#_Toc390256991)

[13 Diagnostics and Exceptions 54](#_Toc390256992)

[13.1 Diagnostic reporting system 54](#_Toc390256993)

[13.2 Exception reporting system 54](#_Toc390256994)

[13.3 Message Descriptions 56](#_Toc390256995)

[13.4 Common Fields 57](#_Toc390256996)

[13.5 Statistics Message Header 57](#_Toc390256997)

[13.8 Camera Exception 63](#_Toc390256998)

[13.9 Camera Exception Flags 64](#_Toc390256999)

[14 Dummy record generation 66](#_Toc390257000)

[15 References 67](#_Toc390257001)

[15.1 Internal References 67](#_Toc390257002)

[15.2 External References 67](#_Toc390257003)

[15.3 Glossary 69](#_Toc390257004)

[16 Appendices 71](#_Toc390257005)

[16.1 Appendix A – Example Initial system configuration 71](#_Toc390257006)

[16.2 Appendix B - Example INSTATION server 77](#_Toc390257007)

[16.3 Appendix C – Analysis of random number generation 82](#_Toc390257008)

[16.4 Appendix D – Excerpts from RFC2104 - HMAC 83](#_Toc390257009)

[16.5 Appendix E - Data structures for session protocol 87](#_Toc390257010)

[16.6 Appendix F – Example Software update procedures 98](#_Toc390257011)

[16.7 Appendix G - Readability of License Plates 99](#_Toc390257012)

[17 VES ADDENDUM - Radar Interface 102](#_Toc390257013)

[17.1 Overview 102](#_Toc390257014)

[17.2 Specification 103](#_Toc390257015)

[17.3 Configuration 104](#_Toc390257016)

[17.4 System hardware and interconnection 106](#_Toc390257017)

[17.5 Usage guidelines 106](#_Toc390257018)

[17.6 Mounting and alignment 107](#_Toc390257019)

[17.7 Data transfer 107](#_Toc390257020)

[17.8 Approvals 109](#_Toc390257021)

[17.9 Error minimisation & correction 110](#_Toc390257022)

[17.10 APPENDIX - Further Guidance notes 113](#_Toc390257023)

# 1 Преглед на документа

Този документ описва пълните спецификации на интерфейса на VES. Някои части от спецификацията все още не са реализирани. Други части може да не отговарят на определени ОЕМ (производител на оригинално оборудване) приложения.

VES системата се предлага в два общи варианта. Базовата VES предлага минимална система без контрол на сесиите, удостоверяване или кодиране. Пълната VES предлага контрол на сесиите, удостоверяване и кодиране. Всяка от версиите може да се поръча с или без софтуер за буфериране на записи върху компактен носител.

Този документ описва широк диапазон от интерфейси и протоколи, които изграждат VES. Малко вероятно е някое приложение да изисква или използва всички описани средства. Системата VES може да се изгради в специализиран вид за определени ОЕМ изисквания.

Консултирайте се подробно с 3М и авторите, за да сте сигурни, че внедряването и конфигурирането за употреба е най-подходящото за дадено приложение.

Този документ ще показва, където е възможно, кои средства липсват в базовата VES. За специфични ОЕМ форми ще има допълнителни забележки.

## 1.1 Въведение

Тази спецификация описва софтуерния интерфейс, който е вграден в устройството 3M System ANPR, което предлага средства за общо таксуване и за изпълнение на нарушения. Тази система е камера 3M ANPR със софтуер, предназначен да предоставя изображение и регистрацията на превозното средство за такса задръствания, за пътна такса и други приложения за изпълнения на нарушения.

Този документ трябва да се чете заедно с съответното ръководство на вашата ANPR камера.

Интерфейсът между Вътрешната станция и Външната ANPR станция е създаден да предлага следните функции:

* Синхронизация на времето на Външната ANPR станция с Вътрешната станция
* Наблюдение на синхронизацията на времето на Външната ANPR станция
* Сваляне на конфигурационни данни за Външната ANPR станция от Вътрешната станция
* Сваляне на софтуер за Външната ANPR станция от Вътрешната станция
* Изпращане от Външната ANPR станция на резюмирани данни за превозните средства индивидуално или по групи
* Изпращане от Външната ANPR станция на снимков материал на превозните средства
* Изпращане от Външната ANPR станция на редовни данни за статуса
* Изпращане от Външната ANPR станция на аларми за статуса
* Превключване на Външната ANPR станция между главна и вторична вътрешна станция

Връзката и потокът с данни на записа с нарушенията (доказателствата) може да бъде с удостоверяване и кодиране. Могат да бъдат взети мерки за защита на целостта на системата за наблюдение, докато записите с нарушения се събират и изпращат.

##### 1.1.1 Физическо конфигуриране

Камерата се нуждае от пусков механизъм, за да започне цикъл за заснемане на регистрационна табела. Този пусков механизъм се генерира вътрешно, когато хардуерът за откриване на регистрационни табели открие присъствие на потенциална регистрационна табела в зрителното поле. Алтернативно, може да се предостави външен пусков механизъм, ако е инсталиран DSRC интерфейс или друго оборудване за пътно наблюдение. Външният пусков механизъм може да бъде хардуерен превключвател или съобщение, изпратено до камерата през етернет или серийна връзка.

При получаването на пусково събитие камерата ще заснеме малък (конфигурируем) брой снимки с предварително избрана стойност на експозицията, докато превозното средство минава през зрителното поле на камерата. В допълнение по относително същото време ще бъде заснета една снимка от общата камера (ако има такава и е включена). От снимките на регистрационната табела камерата ще избере най-подходящите за по-нататъшна обработка. Избраната снимка ще бъде обработена, за да се открие и разчете най-вероятната област, показваща регистрационна табела. Избраната снимка ще има асоциирани дата и час. Ако не бъде намерена регистрационна табела, камерата все пак може да изпрати общата снимка. Когато няма инсталирана отделна обща камера, инфрачервената камера може да заснеме една експозиция в цикъл на генериране на обща снимка. По избор и други подобни снимки могат да бъдат добавени към набора от данни за събитието.

Тези снимки могат да бъдат заснети преди и след главното събитие.

Ако е активирана сесия за изпълнение, камерата изпраща към хост сървъра секретен запис с доказателства, съдържащ всички данни и снимки.

При всички случаи към хост сървъра може да бъде изпратен запис с кратко резюме.

Камерата ALPR е оборудвана с 100baseT Етернет връзка и поддържа TCP/IP V4 протоколи. Камерата ще бъде свързана към VES сървъра и управляващите системи посредством тази Етернет връзка.

VES сървърът ще предостави подходящо приложение, конфигурирано да приема комуникация и пренос на данни от камерата. Камерата поддържа по избор канали за поддръжка, контрол, съобщения, диагностика и изключения.

Хост системата ще изисква механизъм, който да потвърди, че ALPR камерата работи. За да изпълни това изискване, камерата може да бъде конфигурирана периодично да изпраща пулсов сигнал към дадения сървър. Пулсовият сигнал носи информация за оперативния статус. Ако е активирана тази функция, но излъчването е неуспешно, камерата може да бъде конфигурирана да предприеме действия за възстановяване.

Камерата може да бъде конфигурирана да прехвърля всички заснети събития или да прехвърля само тези събития, които са заявени от хост системата, т.е. тя може да работи в режим изпращане на данни или в режим изтегляне на данни. Когато трябва да се прехвърлят всички събития, се препоръчва камерата да изпраща данните към сървъра, тъй като това е най-ефективния метод. Когато са заявени само избрани събития, тогава камерата може да си превключи да използва протокол за изтегляне. Протоколът за изтегляне може да използва съобщения с ASCII или бинарен контрол.

# Общо конфигуриране

#### Това е описано в детайли в ръководството на потребителя към вашата камера.

Всички 372 опции за конфигуриране се настройват или променят посредством команден ред (CLI) или през уеб страница. Командите към CLI могат да произлизат посредством серийна или телнет връзки или посредством скриптови файлове. Системата може да бъде конфигурирана да стартира даден скриптов файл автоматично при стартиране. Този скриптов файл може да бъде запазен локално или да бъде копиран автоматично от FTP сървъра по подразбиране.

Пълният CLI интерфейс е достъпен през:

Серийния порт за отстраняване на грешки

Стандартна ТЕЛНЕТ връзка през порт 23

Серийният порт за отстраняване на грешки ще генерира спонтанни съобщения относно състоянието на системата. Самите генерирани съобщения ще зависят от различни системни настройки и опции за регистриране и отстраняване на грешки. Затова 3М не препоръчва този порт да се използва като машина/машинен интерфейс, а този достъп да е предназначен за конфигуриране на системата и поддръжка на системния софтуер.

372 поддържа дневник на системните събития, който записва ключови състояния на системата и може при необходимост да поддържа лог файл с данни (analyse.log), който проследява заснетите събития. Тези събития може да се преглеждат посредством CLI или да се прехвърлят към хост система посредством ftp.

Следващите раздели разглеждат опциите за конфигуриране на специфично приложение на VES системата. За повече подробности по конфигурирането и работата на вашата ANPR камера, моля, вижте съответното ръководство на потребителя на вашата камера.

## VES специфично конфигуриране

Специфичният курс на работа се избира чрез CLI командата:

>> set route ves

#### Съществува набор от съответни CLI опции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **команда** | **име на параметър** | **с-ност** | **описание** | **бел.** |
| ves show |  |  | показва специфични VES системни параметри |  |
| ves set |  |  |  |  |
|  | host |  | задава ip адрес на хост сървър за хост, който ще приема записи на събития | 1,10 |
|  | alt\_host |  | Ако главният снимков хост не отговаря на пинг заявките или не може да отговори на заявки за отваряне или прехвърляне, тогава системата ще приеме, че той е офлайн и ще превключи прехвърлянията към този алтернативен хост | 10 |
|  | camera | abcd | име или идентификатор на камера. Това е кратко име, а не пълно описание. Дължината е ограничена до 21 символа. Форматът е TBD, но не трябва да включва празно място или запетая. |  |
|  | lane |  | име или идентификатор. Дължината е ограничена до 11 символа. Форматът е TBD, но не трябва да включва празно място или запетая. |  |
|  | store\_host |  | Задава IP адрес на модула SPIKEstore | 14 |
|  | store\_account |  | Задава име на акаунта на модула SPIKEstore | 14 |
|  | store\_password |  | Задава парола за акаунта на модула SPIKEstore | 14 |
|  | store\_path |  | Задава път до папката на модула SPIKEstore | 14 |
|  | directory | /xxx | Директория на локалната система, която се използва за поддръжка на буфер на диска с диагностични файлове и файлове с изключения, когато хост системата е офлайн. Обикновено ще е зададена dos:/ves т.е. поддиректория на флаш картата | 12 |
|  | cabinet |  | ip адрес на кабинетния контролер. Камерата ще пингва този адрес, за да се увери, че връзката към кабинета е цялостна. Задайте този ip адрес да е нула или празен, за да изключите тази функция | 11 |
|  | router |  | ip адрес на мрежовия рутер. Камерата може да пингва този рутер (icmp ping), за да установи, дали локалната мрежа е цялостна. | 11 |
|  | security |  | посочва ниво на сигурност  0x00 – няма (без удостоверяване или кодиране)  0x01 – генерира HMAC на данните с ключ 1 0x02 - генерира HMAC, кодира всички данни с ключ 2  0x04 - генерира уникален ключов блок за всеки ER  0x08 - кодира NVT канал  0x10 – кодира канал за съобщения  0x80 – кодира диагностициране по време на сесия | 7,11 |
| **команда** | **име на параметър** | **с-ност** | **описание** | **бел.** |
|  | crypt |  | (премахнат параметър) Кодирането трябва да е AES256 | 7,11 |
|  | key\_port | nnnn | Портът, на коя система ще слуша за заявка за стартиране на сесия и на коя ще се осъществи обмяната на ключове.  Задайте нула или празно, за да изключите обмяната на ключове. | 11 |
|  | image\_port |  | задава порт на хост система, който ще слуша за връзка за прехвърляне на снимкови записи. Прехвърлянето на снимки е изключено, ако този порт не е зададен | 1,10 |
|  | link\_timeout | nnn | Затваря връзката за снимки или резюмета, ако не са осъществени прехвърляния през зададения брой секунди. Задайте нула, за да изключите. | 2 |
|  | summary\_port |  | Задава порт на хост системата, който ще слуша за връзка за прехвърляне на резюмирани записи. Прехвърлянето на резюмирани записи е изключено, ако този порт не е зададен. | 10 |
|  |  |  |  |  |
|  | nvt port |  | Команден интерфейс за контрол машина-машина между камерата и пътния контролер или вътрешната станция. Камерата слуша през този порт | 10 |
|  | nvt\_timeout | Nn | Изтичане на времето (секунди) на NVT интерфейса. Ако не са получени заявки или пулсови сигнали за зададения брой секунди, контактът се затваря и слуша за друга връзка. |  |
|  | nvt\_mode |  | Избира режим на работа на nvt:  0 – изключен  1 – ASCII интерфейс  2 – бинарен интерфейс |  |
|  | msg port |  | Задава локалния tcp/ip порт, който се използва от канала за съобщения. Камерата слуша през този порт. | 3,10 |
|  | msg\_timeout | nn | Ако не са изпращани съобщения по този канал за съобщения през последните nn секунди, тогава ще бъде изпратено поддържащо съобщение, което да покаже на хост системата, че каналът за съобщения все още е активен. |  |
|  | hb\_port |  | Задава номера на порта за процеса на хост lpr сървъра, който очаква пулсови съобщения. Хост системата слуша през този порт. |  |
|  | heart\_beat | nn | период в секунди между пулсовите съобщения. Ако е зададено 0, няма да бъдат изпращани пулсови съобщения. |  |
| **команда** | **име на параметър** | **с-ност** | **описание** | **бел.** |
|  | not\_alive | nn | ако процесът на пулсиране е в действие, (heart\_beat > 0) тогава, ако пулсът не успее да изпрати съобщението си (т.е. не може да установи връзка) за зададения брой опити, тогава 372 ще опита да започне действие по възстановяване. |  |
|  | trig\_port |  | Портът, на който системата ще очаква TCP/IP пускови съобщения. Камерата слуша през този порт. |  |
|  | trig\_timeout | nn | Ако е отворена връзка към пусков порт, тогава камерата ще изпрати пулсово съобщение на пусковия порт след зададения брой секунди. (Това е прост пулс, който ще покаже на прикачения контролер, че портът е активен.) |  |
|  |  |  |  |  |
|  | transfer | 0xnn | Задава се, за да се избере, кои снимки и съобщения се прехвърлят на сървъра. | 6 |
|  |  |  |  |  |
|  | debug | 0x00 | Това е битово поле, което определя съобщението за отстраняване на грешки, което може да се генерира на серийния порт или stderr порт. Точният формат е TBD |  |
|  | timeout |  | Ако на камерата не е получена снимка на регистрационна табела за зададения брой секунди, тогава в статусния регистър ще бъде зададен един бит. Това може да показва възможна грешка в системата, напр. грешка в камерата, или може да показва, че пътната лента е затворена. |  |
|  | ping\_interval |  | Интервал, през който фонов процес наблюдава наличие/липса на сървъри |  |
|  | ping\_wait | nn | Брой милисекунди за изчакване на пинг отговор преди да се реши, че сървърите са офлайн. |  |
|  | nlog |  | Максимален брой снимки, които да се буферират на флаш диск, след което най-старата снимка ще бъде презаписана. | 12 |
|  | file\_size |  | Максимален размер (в килобайти) от снимковия набор от данни, който да се буферира на флаш диск. Този параметър трябва да бъде избран внимателно с цел nlog да оползотвори най-добре наличното място. Доказателствени записи с по-голям от зададения размер ще бъдат изчистени. | 12 |
|  | ack\_enable |  | Задава битове, за да задължи хост системата да потвърждава всяка заявка за прехвърляне. | 8 |
|  | ack\_timeout | nn | Брой милисекунди за изчакване на потвърждение преди автоматично отново да се изпрати даден запис |  |
|  | ack\_retry | nn | брой опити, които системата ще направи да изпрати даден запис преди да се откаже и да го изчисти. |  |
| **команда** | **име на параметър** | **с-ност** | **описание** | **бел.** |
|  | scan |  | Брой секунди между сканиранията на флаш диск буфера, за да се провери, дали има файлове, чакащи прехвърляне на хост системата |  |
|  | threshold |  | Снимковите събития се изчистват, ако достоверността на разчитане на регистрационни табели е под това ниво |  |
|  |  |  |  |  |
|  | orientation | n | посочва ориентацията на камерата като кардинална точка, където 0-N, 1-NW, 2-W и т.н. |  |
|  | time\_slot | nn | Диагностичното докладване се отклонява от номиналния период със зададения процент | 9 |
|  | context\_count |  | необходим брой поддържащи допълнителни снимки (макс. 8 – по подразбиране 0) | 11 |
|  | context\_offset |  | Номинално време между допълнителните снимки (милисекунди). Виж също capture параметър ctx\_sample | 11 |
|  | context\_order | n | Задава реда на допълнителните снимки. Снимките могат да са:  0 – разделени по равно преди/след пусковия момент  1 – всички са преди пусковия момент  2 – всички са след пусковия момент  За опция нула винаги ще има четен брой допълнителни снимки. Опции 1, 2 позволяват нечетен брой допълнителни снимки | 11 |
|  | slot\_age\_limit | n | След като даден запис е бил в паметта през зададения период (секунди) без успешно прехвърляне към хост системата, той или ще бъде изтрит, или прехвърлен на компактния флаш носител (ако се събере) |  |
|  | cf\_age\_limit | nn | Задава максималния брой часове за запазване на данни в компактната флаш памет. Всички данни по-стари от това ще бъдат изтрити и няма да бъдат прехвърлени на хост системата. |  |
|  | summary\_rate | nn | Задава контрол върху прехвърлянето на резюмирани записи, основан на броя висящи доказателствени записи. Задайте нула, за да изключите този контрол. Минимална активна стойност: 10. Когато е зададена, ако има повече висящи доказателствени записи от този брой, прехвърлянето на резюмирани записи от хранилището ще бъде преустановено. Прилага се хистерезис от 10%. |  |
|  | radar\_enable | n | Задайте 1, за да включите интерфейса на детектора за скорост | 13 |
|  |  |  |  |  |
|  | radar\_threshold | nn | Задава праг на измерване на детектора за скорост, под който записите ще бъдат изчиствани. | 13 |
| **команда** | **име на параметър** | **с-ност** | **описание** | **бел.** |
|  | store\_enable | b | Задайте 1, за да включите прехвърлянето на данни към модула SPIKEstore (ако е наличен и конфигуриран) |  |
|  | max\_batch\_size | nn | Само Пакетно Резюме. Когато текущият пакет достигне този размер, пакетът е завършен и нареден за прехвърляне |  |
|  | bsr\_port |  | Портът, към който се насочват пакетните резюмирани записи |  |
|  | bsr\_nlog |  | Броят пакети от резюмирани записи, които са запазени. Когато е достигнат този брой, най-стария пакет се презаписва. |  |
|  | bsr\_interval |  | Интервалът на обработване на пакети (секунди), след който се стартира нов пакет |  |
| ves send | nnnnn fffff |  | прехвърляне на доказателствен запис с event\_id nnnn и file\_id ffff, където  nnnn е id, генерирано в шестнайсетична последователност от камерата, за всяко събитие и отбелязано в резюмирания запис.  fffff е шестнайсетичното file\_id, отбелязано в резюмирания запис |  |
|  |  |  |  |  |
| ves del | nnnnn fffff |  | изтрива доказателствен запис с event\_id nnnn и file\_id fffff, където  nnnn е id, генерирано в шестнайсетична последователност от камерата, за всяко събитие и отбелязано в резюмирания запис.  fffff е шестнайсетичното file\_id, отбелязано в резюмирания запис |  |
| ves diag | show |  | показва текущата диагностична конфигурация – виж специфичния раздел по-долу |  |
|  | set |  | задава специфичен диагностичен параметър – виж специфичния раздел по-долу |  |
| ves excep | show |  | показва текущата конфигерация за изключения – виж специфичния раздел по-долу |  |
|  | set |  | задава специфичен параметър за изключения виж специфичния раздел по-долу |  |
| ves stat |  |  | Показва статистика и хистограми, за да покаже латентности при снимките |  |
| ves list |  |  | Показва списък и статус на снимковите събития в ram |  |
| ves loc |  |  | задава форма за подробно описание на местоположението на камерата |  |
| ves secret |  |  | задава споделена тайна (или парола), ако е необходимо. Форматът е свободен текст с максимална дължина 64 символа. Виж по-долу за подробности | 11 |
| **команда** | **име на параметър** | **с-ност** | **описание** | **бел.** |
| ves init |  |  | Инициализира или реинициализира файловата система. Заснемането на снимкови събития е преустановено, всички файлове в системата се изтирват, след това се разпределя нов пълен набор от файлове. Тази дейност може да отнеме десет минути за изпълнение на система с устройство от 1 гигабайт. |  |
| ves  remove |  |  | Премахва всички записи от флаш носителя |  |

забележки

1. Това обозначава IP адреса на хост сървъра, който ще приеме данните за регистрационната табела и снимките.
2. Веднъж щом камерата е отворила връзка за прехвърляне на снимки, тя ще задържи връзката за бъдещи прехвърляния. Ако няма прехвърляния в рамките на зададения брой секунди, тогава връзката ще бъде затворена
3. изтрито
4. изтрито
5. изтрито

За всяка конфигурация трябва съответно да бъдат зададени други параметри на системната конфигурация. Този параметър контролира, как се прехвърлят файловете към хост системата

|  |  |  |
| --- | --- | --- |
| 0x0001 | прехвърля регистрационна табела |  |
| 0x0002 | прехвърля цяла снимка на регистрационна табела |  |
| 0x0004 | прехвърля предна обща снимка |  |
| 0x0008 | прехвърля допълнителни снимки |  |
| 0x0010 | Изисква пълно изтриване |  |
| 0x0020 | добавя текст към регистрационна табела и инфрачервени снимки |  |
| 0x0040 | изчиства събития под прага |  |
| 0x0080 | прехвърля събития само при заявка от DSRC или вътрешната станция (т.е. избира протокол за изтегляне) |  |
| 0x0100 | при неуспех на tx буферира на диска |  |
| 0x0200 | Включва прехвърляне на M001 съобщение |  |
| 0x0400 | Включва прехвърляне на M002 съобщение |  |
| 0x0800 | Включва прехвърляне на предупредително съобщение |  |
| 0x1000 | Включва прехвърляне на съобщение за грешка |  |
| 0x2000 | Прехвърля пулс като H001 съобщение |  |
| 0x4000 | Прехвърля съобщение за състоянието на сесията |  |
| 0x8000 | Включва прехвърляне на резюмирани записи |  |
| 0x10000 | Включва създаване на кратки резюмирани записи |  |
| 0x20000 | Включва прехвърляне на пакетни резюмирани записи |  |

#### Прехвърляният снимков файл има прикачен SHA1 HMAC удостоверителен запис. Удостоверяващото поле или целият запис (с изключение на първия ред от хедъра) могат да бъдат кодирани по желание. Може да бъде избрана дължината на кодиращия ключ. AES може да бъде 128 или 256 битов. Подразбира се, че key\_1 и key\_2 ще бъдат със същата дължина. Ключове 1, 2 се прилагат като част от стартовия протокол на сесията. OEM конфигурация може да премахне тези опции и да наложи конфигуриране в специфичен режим.

1. Прехвърляния по каналите за снимков и резюмиран запис могат да бъдат потвърждавани. Задава битове, в зависимост от това, кои канали изискват потвърждението:

|  |  |
| --- | --- |
| 0x01 | изисква потвърждение за прехвърляне на снимков запис |
| 0x02 | изисква потвърждение за прехвърляне на резюмиран запис |
| 0x04 | изисква потвърждение за диагностични записи |
| 0x08 | изисква потвърждение за записи с изключения |
| 0x10 | Изисква потвърждение за прехвърляне на spikestore записи |
| 0x20 | Изисква потвърждение за прехвърляне на пакетни резюмирани записи |

1. При множество камери с външни станции, докладващи редовни диагностики е необходимо да се вземат мерки да се гарантира, че тези редовни комуникации от външните станции са отложени във времето. Този параметър задава процентното изместване на комуникацията на това местоположение с вътрешната станция. По този начин, примерно, ако станцията трябва да изпраща диагностичните си данни на всеки 5 минути и това изместване е настроено на 40 тогава това местоположение ще прехвърля данните на всеки 5 мин. + 2 мин.
2. Ако някоя от тези конфигурационни опции бъде променена, системата може да поиска рестартиране преди промяната да влезе в сила. (Особено, ако има отворена връзка, тогава новият IP адрес не може да влезе в сила, докато или връзката се затвори при естествено намалял трафик, или връзката се поднови в резултат на рестартиране.)
3. Тази опция няма да се вижда на базовите VES системи
4. Тази опция няма да е включена на система, която няма инсталирана поддръжка на компактен Флаш носител.
5. Опции, които са валидни и налични само на системи, конфигурирани да поддържат радарния интерфейс. Виж приложението за доплеров радар към този документ.
6. Опции, които са валидни и налични само на системи, конфигурирани да поддържат модула за разпределяне крайпътните запаметяващи устройства SPIKEstore. Виж специфичното SPIKEstore ръководство и приложението към този документ.

## Идентификация на местоположението

Съществуват няколко различни идентификатора за местоположение. Съществува дълго описание на местоположението, посочващо със смислени за потребителя термини разположението на камерата. Този низ се докладва като част от резюмирания запис и доказателствения запис, но по друг начин е неизползваем. Може да бъде добавен към снимките, но е TBD.

Всяка камера има отделен идинтификатор на камерата и идентификатор на пътната лента. Това са кратки низове в свободен формат (които не трябва да включват празно място и запетая). Те обикновено са „машинни“ идентификатори.

Заявките извикват идентификатори за инфрачервена камера и допълнителна камера. Тъй като камерата е само едно устройство, само един идентификатор е предоставен да изпълни и двете нужди.

Навсякъде в този документ има препратки към site\_id. Той се образува от свързването на идентификатора на камерата и идентификатора на пътната лента с долно тире помежду им, напр. камера С123 и лента 2 ще имат идентификатор на местоположението С123\_2.

1.1 Филтриране на дубликати

Системата може да бъде настроена да отхвърля дублирани регистрационни табели, получени на последователни или около последователни събития. Филтрирането на дубликати може да се изключва или включва на избрани нива.

|  |  |  |
| --- | --- | --- |
| cap set duplicates | n | където n  0 – отказ на дубликати изключено   1. – отказ на дубликати включено. Сравняването за „еднакви регистрационни табели“ ще счита много близки символи за еднакви символи (напр.: В се счита за еднакво с 8) 2. – отказ на дубликати включено. Съпоставянето използва общо откриване на подниз за подпомагане на откриването на дубликати, когато една снимка може да има липсваща част (напр. при закриване от друго превозно средство или пешеходец) |
| cap set duplicate\_level | n | определя процентовото съвпадение между регистрационни табели, които се считат за дубликати. |
| cap set duplicate\_depth | n | Брой редове в списъка с историята на дубликатите. |
| cap set duplicate\_age | n | Максимална възраст (в секунди) на всеки ред в списъка с историята на дубликатите. Ако дадена регистрационна табела се появи отново преди да е изтекло времето на реда й, възрастта му се подновява от новото събитие. |

Историята на дубликатите има историческа дълбочина от duplicate\_depth регистрационни табели. Регистрационните табели в списъка с историята ще бъдат изчистени, ако не са се появили отново в продължение на duplicate\_age секунди.

## Синхронизация на времето

Това приложение има изискване регистрационните табели, заснети от камера да се съпоставят със събития открити от други системи. Това се получава като се отбелязва времето на снимките. За да работи това, двете системи трябва да си поделят общ часовник.

Системата на камерата ще се опита да поддържа локално време и дата посредством връзка с BSD / UNIX style SNTP (за предпочитане) или времеви сървър на определена хост машина през порт 13. Повечето Unix и NT системи ще предоставят тази услуга или ще качат услуга, която да го направи. Извикването на тази услуга може да бъде изключено като се настрои IP адреса на хоста за времевата услуга на 0. Ако тази услуга е изключена, тогава вътрешния часовник / календар на 372 ще поеме в съответствие с външното време. Забележете също, че времето, което се отбелязва на съобщенията в дневника със събития, се взема от календара.

Локалното време може също да се прикачи към GPS времеви приемник, който предоставя NMEA записи и PPS (пулс на секунда) сигнал. Когато се направи това, хардуерното внасяне вече не е приложимо.

При необходимост 3М може да препоръча SNTP сървър, подходящ за употреба с  
Windows NT или Win2k.

Ако използва съвместим SNTP сървър всички времена ще бъдат UTC. Ако се използва времеви сървър, тогава всички времена ще бъдат настроени, като се използва стандарта, който е настроен на времевия сървър.

Подробности по синхронизация на времето могат да се намерят в главния конфигурационен документ.

NTP и SNTP протоколите са документирани в *RFC1305, RFC1769* и *RFC2030.*

Времевият протокол е описан в *RFC0867*. В това приложение времевият клиент ще очаква времето във формат:

Day Mon dd hh:mm:ss Year

(Времевият протокол не се препоръчва, за предпочитане е SNTP.)

Управлението на времевия сървър са в системната страница на CLI или на уеб интерфейса. Съответните механизми са:

|  |  |  |
| --- | --- | --- |
| **Команда** | **Параметър** | **Описание** |
| system set time\_server | aa.bb.cc.dd | Посочва IP адреса на сървъра, който да се използва при заявките в интернет за „времеви клиент“ и SNTP.  Задайте 0, за да изключите повикванията към времеви сървър |
| system set alt\_time\_server | aa.bb.cc.dd | Задава IP адреса на алтернативния времеви сървър, с който ще се контактува, ако главния не отговаря. |
| system set sntp\_latency | nn | Задава приемлива латентност на връзката в най-лошия случай на nn (десетично) милисекунди.  Ако, след като е изпратена sntp заявка и и е получен отговор, показваната латентност надвишава този лимит, тогава данните се изчистват и се прави нов опит. |
| system set sntp\_window |  | Задава прозорец, в рамките на който времето трябва да се установи (милисекунди), т.е. ако е зададено 200 мсек. системата ще се опита да поддържа номинално установяване на времето в рамките на 200 мсек. |
| system set time\_poll | nnn | Интервал в секунди между синхронизациите на времето. |
| system set sntp\_max | nn | Ако вътрешното време и времето на хоста се различават с повече от nn часа (по подразбиране 24), тогава може да съществува сериозна грешка евентуално с времевия хост сървър. Системата няма да опита да коригира времето. Задайте този параметър 0, за да стартирате винаги процеса на корекция на времето. |
| System set sntp enable | 0xnn | Задава включване на sntp и времеви.  По подразбиране е 0x13.  0x01 – включва sntp клиент  0x02 – включва времеви клиент  0x04 – включва GPS клиент  0x10 – включва sntp сървър |

На времевите корекции се прилага нискостепенно филтриране. Ако резултатът от времева корекция е все още извън sntp\_window, тогава се изпълнява нова времева корекция. След 20 опита системата ще преустанови за time\_poll преди да опита отново. Първоначалното установяване на времето след стартиране на системата, когато вътрешния часовник може да е поел за няколко секунди, може да отнеме няколко sntp опита. Ако времевото несъответствие е минути или часове, тогава няколко time\_poll периода може да изтекат преди се постигне пълно установяване.

Ако system set sntp\_max 0, тогава винаги се прилага първоначалната времева корекция без нискостепенно филтриране и необходимост от каквато и да е корекция. Това позволява на системата бързо да постигне статус на синхронизиране при стартирането на системата.

Веднъж щом камерата е постигнала установяване на времето, алармата, отбелязваща загуба на установяване на времето, няма да се включва, докато не изтече период от 20 минути без успешно SNTP обновяване.

Камерата може също да функциаонира като SNTP времеви сървър. Камерата ще връща „несинхронизирана“ на всеки клиент, докато не бъде постигнато успешно установяване.

Информация за позицията

Когато камера е оборудвана с GPS приемник, може да се записва и прехвърля информация за позицията като част от Резюмираните и Доказателствените записи. Виж подробности по конфигурирането на GPS (документирано на друго място) за правилното конфигуриране на GPS подсистемата.

## Валидизация на системната конфигурация

(Виж съответното ръководство на потребителя за ANPR камера за подробности)

Камерата ще използва активни копия на конфигурационните данни, запазени в SRAM. Въпреки това при продължителна липса на захранване (повече от седем дни), тези данни могат да бъдат компрометирани. При такива обстоятелства активните копия ще бъдат опреснени с копия, пазени във флаш копието на „boot“ паметта. Съществува процедура за подновяване на флаш копието на „boot“ паметта. Като част от процеса на валидизация на системата, конфигурационните детайли на „boot“ флаш паметта и на SRAM трябва да са еднакви и освен това съответното CRC на тези данни и програмните снимкови файлове трябва също да съвпадат. Командите, използвани за управление на системата са:

|  |  |
| --- | --- |
| ***Команда*** | ***Описание*** |
| keyfile update | копира ключовите конфигурационни файлове от sram на „boot“ флаш паметта |
| keyfile restore | възстановява текущия набор от ключови конфигурационни файлове от флаш копията на „boot“ паметта в работни sram копия |
| keyfile validate | Гарантира, че CRC на ключовите конфигурационни файлове в „boot“ флаш паметта и sram съвпадат и ако съвпадат, запазва съответно CRC в eprom. Тази команда изисква код за сигурност, за да се изпълни. |
| keyfile show | Докладва контролната сума на eprom и CRC на конфигурацията и на приложението |

Съществуват два набора от файлове:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| приложение | главно изображение на приложението  изображение на гейт масива  шрифтови файлове  anpr файл |  |
| набор от конфигурационни файлове | конфигурационен файл  ксписък за контрол на достъпа |  |

Съществуват процедури за автоматично прехвърляне и обновяване на системни файлове. Те са описани в приложение към този документ.

# Последователност на събитията

#### Това е кратко описание на поведението на системата

Системата на камерата може да прехвърля данни или директно към хост системата (протокол на изпращане) или само при заявка от хост системата (протокол на изтегляне). Когато повечето от записите трябва да бъдат прехвърлени на хост системата, тогава се препоръчва използването на архитектурата за изпращане, тъй като би била най-ефективна. Архитектурата за изтегляне помага да се намали потока от данни, ако получаващата страна прилага недублиране и изисква само определени записи.

VES системата обикновено е оборудвана с непроменливо запаметяващо устройство, където ще се извършва локално буфериране, особено за система на изпращане, където заявката за изпращане може да бъде забавена. Това обикновено би представлявало компактно флаш устройство между 64 Мб – 3 Гб.

## Протокол на изпращане

* + 1. Пусков сигнал се получава или от външен пусков вход (сериен или етернет като например от DSRC система), или от вътрешен хардуер за откриване на регистрационни табели. Етернет пусковите механизми се препоръчват само за системи с много бавен трафик.
    2. Камерата започва да търси регистрационна табела на превозно средство. Камерата продължава да търси табела, докато изтече „отворения“ период на пусковия механизъм или докато превозното средство напусне зрителното поле. Отвореният период по подразбиране е около 150 мсек, но е напълно конфигурируем.
    3. При първа възможност по време на експозицията камерата заснема обща снимка (ако тази опция е включена). Ако е заявено, се заснемат и други общи снимки, докато превозните средства преминават през зрителното поле.
    4. В края на отворения период или след преминаването на превозните средства ANPR обработва най-добрата открита снимка на регистрационна табела.
    5. Създават се резюмирани и (ако е заявено) доказателствени записи, които съдържат посочените снимки и други данни за събитието със съответното удостоверяване и кодиране. Този запис се запазва във вътрешна таблица с място за 100 събития. Ако тази таблица се напълни, най-старият запис се презаписва или, ако е включено буфериране, може да се прехвърли към непроменливото запаметяващо устройство.
    6. Тогава пълният запис на събитието се прехвърля към хост системата. Ако няма връзка с хост сървъра, записът се буферира в непроменливото запаметяващо устройство, докато бъде или успешно прехвърлено, или презаписано, когато мястото за съхранение се напълни.
    7. При завършване на прехвърлянето на събитието може да се генерира М004 съобщение.

## Протокол на изтегляне

* + 1. Пусков сигнал се получава или от външен пусков вход (сериен или етернет като например от DSRC система), или от вътрешен хардуер за откриване на регистрационни табели. Етернет пусковите механизми се препоръчват само за системи с много бавен трафик.
    2. Камерата започва да търси регистрационна табела на превозно средство. Камерата продължава да търси табела, докато изтече „отворения“ период на пусковия механизъм или докато превозното средство напусне зрителното поле. Отвореният период по подразбиране е около 150 мсек, но е напълно конфигурируем.
    3. При първа възможност по време на експозицията камерата заснема обща снимка (ако тази опция е включена). Ако е заявено, се заснемат и други общи снимки, докато превозните средства преминават през зрителното поле.
    4. В края на отворения период или след преминаването на превозните средства ANPR обработва най-добрата открита снимка на регистрационна табела.
    5. При разчитане на регистрационната табела се приготвя Резюмиран и Доказателствен запис с общ идентификатор.
    6. Резюмираният запис се нарежда за изпращане към Вътрешната станция възможно най-скоро. Остарелите Резюмирани записи ще се съхраняват на непроменливо запаметяващо устройство.
    7. Вътрешната станция ще потвърди получаването на Резюмирания запис. При положително потвърждение той ще бъде отбелязан за изтриване, а при отрицателно потвърждение той ще бъде изпратен повторно.
    8. Вътрешната станция ще реши, дали иска или не, Доказателствения запис да бъде запазен. Ако реши да бъде запазен, ще изпрати заявка, като използва общия идентификатор, в противен случай ще заяви изтриване, като използва общия идентификатор.
    9. По желание Вътрешната станция ще потвърди получаването на Доказателствения запис. В случай на неуспешно потвърждение, записът ще бъде изпратен повторно.
    10. При успешно потвърждение Доказателствения запис, запазен в паметта, ще бъде отбелязан, като прехвърлен.
    11. По желание Вътрешната станция може напълно да изтрие даден запис, който е бил успешно получен.
    12. Когато броят на Доказателствените записи, запазени в паметта наближи максимално допустимото, те ще бъдат изтрити, ако са отбелязани с положително потвърждение за прехвърляне или запазени в непроменливо запаметяващо устройство, ако не е било получено такова потвърждение.

## Локално запаметяващо устройство за доказателствени записи

Целта е Доказателствените записи да се съхраняват само в непроменливо запаметяващо устройство, когато не е възможно нормално обработване, поради фактори като неуспешна комуникация или грешка във Вътрешната станция. Броят на Доказателствените записи, запазени в паметта може да бъде не повече от 200, в зависимост от размера на записа и параметрите им.

Докато даден Доказателствен запис се съхранява в RAM, може непрекъснато да се правят заявки за прехвърляне. Обаче, ако веднъж е прехвърлен към непроменливо запаметяващо устройство, записът ще бъде изтрит след първото успешно прехвърляне.

Вътрешната станция може да използва команди за пълно изтриване, за да премахне даден запис, щом веднъж е бил успешно получен. Употребата на такива команди може да премахне нуждата от потвърждение за прехвърляне на Доказателствен запис. В такъв случай системата трябва да бъде конфигурирана да не изисква потвърждение за прехвърляния на Доказателствени записи.

Общият идентификатор се състои от две числа. Това са *event\_id* и *file\_id*.

*Еvent\_id* е нарастващо, число създавано от камерата за всяко събитие. Това е число е в рамките на 2^32.

*File\_i*, което също е нарастващо число, ще определя мястото на съхранение в камерата (името на файла е основано на това число). Рамките на това число зависят от капацитета на запаметяващото устройство. Така например при система със запаметяващо устройство от 1 Гб и максимален размер на файла с доказателствения запис от 128к, ще бъде в рамките на около 7500.

## Допълнителни снимки

При работа на SPIKE+ платформа, оборудвана с FULL VES, системата VES има способността да заснема допълнителни снимки освен главните общи снимки, заснети по време на събитието. Тези допълнителни снимки се заснемат:

* всички преди главното събитие
* всички след главното събитие
* или разпределени по равно преди и след главното събитие

Може да бъде посочен интервалът между допълнителните снимки. Заявеният интервал е ограничен от честотата на дискретизация. По подразбиране честотата на дискретизация е настроена на 2, т.е. допълнителна снимка се запазва в камерата на всеки 40 мсек и затова стойността на допълнителните интервали е 40 мсек.

## Данни за траекторията

Камерата SPIKE има способността да изчислява и докладва траекторията на регистрационна табела, която минава през зрителното поле на камерата. Докладът за траекторията се състои от две двойки координати x1,y1 и x2,y2 заедно с времеви интервал. Първата двойка координати представлява приблизителния център на регистрационната табела на първото разчетено местоположение, което е докладвано със заснето изображение и първото отбелязано време. Втората двойка координати представлява резултата от второто разчетено местоположение на регистрационната табела. Интервалът е времевия интервал между първото и второто разчитане. Втората позиция на регистрационната табела може да се окаже преди или след първото разчитане, както пространствено, така и хронологично.

Данните за траекторията биха били точни и смислени, единствено ако камерата е конфигурирана да генерира тези данни като събира няколко снимки, докато регистрационната табела минава през зрителното поле. За да включите това, конфигурирайте системата:

capture set list\_length n

където за n е зададен броят снимки, запазени за анализ. По подразбиране това ще е настроено на 1. Данните за траекторията ще се базират на хардуерното откриване на регистрационни табели и поради това ще са доста приблизителни. За точни траектории при бърз трафик подходяща стойност би била някъде между 5-10. За по-бавен трафик 15 може да бъде по-подходящо.

Имайте предвид, че съществуват някои ограничения, когато се изчисляват траектории.

Ако дадено превозно средство е неподвижно, не може да се изчисли никаква траектория. Ако дадено превозно средство се движи много бавно, тогава може движението да е твърде незначително, за да достигне подходяща траектория преди да бъде създаден доклад за регистрационна табела. Ако много бавният трафик е нещо нормално за определено местоположение, тогава може да е от полза да се увеличи параметъра count на capture.

Ако дадено превозно средство се движи много бързо, тогава може да се направи само една използваема експозиция и траекторията не може да бъде изчислена.

Ако дадено превозно средство се движи диагонално, така че регистрационната табела е изцяло в зрителното поле за съвсем кратко разстояние, тогава разликата между такива снимки може да бъде много малка и като резултат от това всяка траектория може да има твърде малка база за изчисление.

## Ръководство за очаквани размери на файлове

Таблицата по-долу показва някои изчислени размери на файлове, основани на jpg фактор 50 за снимки, заснети при условия на облачност. Ако снимките са заснети при ярка слънчева светлина ще съдържат повече детайли и следователно ще имат по-големи файлове.

|  |  |  |
| --- | --- | --- |
| **избран метод на прехвърляне** | **описание** | **изчислен размер на файла** |
|  | файл само на табела | 2k – 4k |
|  | табела плюс пълна ИЧ | 8k – 24k |
|  | табела плюс обща | 30k – 50k |
|  | табела плюс пълна ИЧ плюс обща | 34k – 60k |
|  | табела с вграден текст | 4k – 8k |
|  | табела плюс обща, и двете с вграден текст | 33k – 50k |
|  | Табела, ИЧ, обща и две допълнителни снимки. Всички снимки с вграден текст. | 115k |

Важно:

* + 1. Ярката слънчева светлина може да увеличи размера на пълната ИЧ снимка с около 10k
    2. 850 nm светлина може също да увеличи фоновите детайли в пълната ИЧ снимка, като увеличи размера с още 5k

Ако се използва компактна флаш памет или твърд диск, тогава мястото за съхранение може да се увеличи до 2 Гб максимум за файлова система FAT16. В този случай трябва да се вземат предвид максималния файлов лимит на системата FAT16, която се ползва на камерата, 32к файла, както и очакваните размери на файловете със снимкови събития. Внимание: на 2 Гбайта FAT16 файлова система, мястото определено за отделни файлове нараства със стъпки от 32 кбайта.

За ефикасност и надеждност на запаметяващото устройство свободното място се преизчислява за всеки файл. Това ще става автоматично при стартиране на системата, ако файловата система не съществува. Или мястото за съхранение на файлове може да бъде изчистено и преместено посредством init командата на VES. За да избегнете грешки и повреди в системата по време на работа, тази команда изисква въвеждането на системен код за сигурност.

* 1. Изчисления на капацитета на системата

Трафикът на връзката с хоста обикновено е предназначен да се справи с приблизително двойно по-голямо натоварване от най-тежкия случай, т.е. изчислете най-големия файлов размер при едно събитие в секунда.

Камерата изпраща няколко потока с данни към хост системата. Потоците включват резюмирани записи, доказателствени записи, пулсови записи, записи от порта за съобщения, диагностики и изключения. При значително натоварен трафик доказателственият запис ще обхване над 90% от общия поток, така че при първоначалните изчисления за управление на трафика всички останали данни могат да бъдат игнорирани.

Доказателствените записи могат да бъдат извлечени както от сигнала на живо, така и от данни, запазени на CF диск по време на прекъсване на комуникациите или вътрешната станция. Потокът от сигнала на живо е ограничен по естествен път от потока на трафика. Потокът от компактната флаш памет е ограничен да лимитира общата заявка за трафик до около един запис в секунда. Данните от този източник се препращат към хост системата през втора връзка, която се осъществява по нископриоритетен линк. Сигналът на живо винаги ще има предимство.

Освен това големи количества данни могат да се извличат от интерфейса за стрийминг на изображения (визьор). Този интерфейс също ограничава потока от данни, а) за да гарантира, че комуникационният трафик не е „потопен“ и b) за да гарантира, че нормалното функциониране на ANPR не е компрометирано.

# Стрийминг интерфейс

Данните се прехвърлят към вътрешната станция през стрийминг интерфейс с TCP/IP сокет. Този раздел описва прехвърлянето през стрийминг TCP/IP интерфейса. Съществуват два потока данни, пълните записи с доказателствени снимки и по-кратките резюмирани записи. Всеки от тези потоци по-нататък се разделя на жив / високоприоритетен поток данни и нископриоритетен поток данни, който пренася записи от офлайн запаметяващо устройство.

## Снимков запис

При настъпване на снимково събитие една или повече от снимките могат да бъдат запазени за прехвърляне. Това са областите с регистрационни табели, пълните снимки, съдържащи регистрационни табели, обща снимка и вторичните допълнителни снимки ( ако съответните опции са включени). Снимките са комбинирани в един запис, към които е прикрепен HMAC удостоверяващ запис.

Системата ще се опита да отвори връзка към снимковия хост. Ако не успее и буферингът не е включен, данните ще бъдат изчистени. Ако буферингът е включен, данните се запазват на диск за прехвърляне по-късно, когато връзката може да бъде възстановена.

Ако няма налични записи за прехвърляне, връзката се запазва отворена за link\_timeout секунди преди да се затвори.

Форматът на записа включва хедър в ASCII текст, следван от блоб (голям бинарен обект) с доказателствения запис със снимкови данни. Хедърът позволява снимковите данни да бъдат идентифицирани, проверени за пълнота и запазени на хост системата без неотложна необходимост да се разкодира блоба.

Структурите от подробни данни във формата на записа могат да бъдат намерени в приложение към този документ. Записите в този текст са само за илюстриране и може да не отразяват точно продуктовия софтуер. При всички случаи софтуерните разработчиците трябва да имат предвид продуктовите хедър файлове.

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];

|  |  |  |
| --- | --- | --- |
| **Поле в хедъра** | **Описание** | **Бележки** |
| MMMM | Магическо число от 4 символа, показващо а) начало на запис на снимково събитие, b) версия на данните на снимково събитие |  |
| SSSSS | Идентификатор на местоположението – променлива дължина от макс. 30 символа – формиран от свързването на id на камерата и id на пътната лента (виж по-горе) |  |
| sssss | 8 шестнайсетични символа (0 подравнени наляво малки символи) Номер на текуща сесия |  |
| QQQQQ | 8 шестнайсетични символа (0 подравнени наляво малки символи) представяне на последователен номер на събитието. Този номер може да се използва като препратка към NVT интерфейса, за да идентифицира доказателствения запис за прехвърляне. |  |
| UUUUU | 8 шестнайсетични символа, времето на заснемане на изображението в UTC (секунди) | 1 |
| mmm | 3 шестнайсетични символа, времето на заснемане на изображението в UTC (мсек) | 1 |
| LLLLLL | 6 шестнайсетични символа, цялата дължина на блоба от данни (включително всички байтове. Блобът започва с въпросния случай и завършва с резюме) |  |
| CCCC | 4 шестнайсетични символа, crc16 на блоба с данни. Това crc позволява процесът на получаване да потвърди, че блобът е пристигнал цял без нужда от незабавно декодиране и повторно удостоверяване. |  |
| ididid | осем шестнайсетични символа (променлива дължина) идентификатор на името на файла. Този идентификатор се използва като препратка към записите, запазени на компактна флаш памет. |  |
| \n |  |  |
|  | *Всичко посочено по-горе в чист текст – всички следващи данни могат да бъдат кодирани.* |  |

* + 1. Времето, отбелязано на снимката, ще бъде времето на заснемане на първата обща снимка, ако има такава. Ако няма обща снимка, тогава ще бъде използвано времето на ИЧ снимка.
    2. Резюмирания запис, включващ ves\_image\_data се изпраща в чист текст. За да се избегне използването му като средство за атака на кодировката на разпознаващия запис, на последният се добавя представка от 32 байта произволно число.
    3. Процесът на AES кодиране използва блока от данни, като представлява кратно на 32 байта. Добавя се подравняване, за да се гарантира, че това е дадения случай. От това следва, че удостоверяващият блок е винаги последните 20 байта от блоба. Винаги ще има поне един байт подравняване. Този байт ще съдържа дължината на подравняването. Зареждат се последващи байтове с нарастваща стойност. Така, ако са необходими пет байта подравняване, те ще имат стойности 5,0,1,2,3

Внимание: Routines, които правят разбор на данните в хедъра, трябва да имат предвид, че допълнителни полета могат да бъдат добавяни към текстовия ред или като диагностика, или като разширени протоколи.

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.

Ако са включени потвържденията, системата ще очаква прехвърлянето на записа да бъде потвърдено. Следващи записи няма да бъдат прехвърляни по даден канал, докато записът не бъде успешно потвърден или докато не бъдат осъществени всички повторни опити.

###### Всички бинарни данни се предават в network byte order

По желание системата може да бъде конфигурирана да използва уникална двойка ключове за всеки доказателствен запис. Ако системата е конфигурирана в този режим, магическото число прехвърлено в хедъра показва, че това е въпросния запис и ключов блок предхожда блоба.

Ключовият блок включва:

typedef packed struct ves\_key\_block

{

byte nonce[16]; byte reserved[16]; byte ukey\_1[32]; byte ukey\_2[32];

} VKB;

Софтуерът на камерата генерира ukey\_1 и ukey\_2. Ключовият блок е кодиран с key\_2, получен като част от контролния протокол на сесията.

Ключовете ukey\_1 и ukey\_2 се използват да удостоверят и кодират оставащата част от Доказателствения запис.

Не се прилага локално CRC или удостоверяване като част от ключовия блок. Това е умишлено.

## Резюмиран запис

По желание системата може да препраща резюмиран запис за всяко записано събитие. Резюмето включва:

MMMM SSSSS ssssss QQQQQQQQ UUUUUUU mmm LLLLLL ididid\n packed struct ves\_image\_data

Където

|  |  |  |
| --- | --- | --- |
| **Поле в хедъра** | **Описание** | **Бележки** |
| MMMM | Магическо число от 4 символа, показващо а) начало на запис на снимково събитие, b) версия на данните на снимково събитие |  |
| SSSSS | Идентификатор на местоположението – променлива дължина от макс. 30 символа – формиран от свързването на id на камерата и id на пътната лента (виж по-горе) |  |
| sssss | 8 шестнайсетични символа (0 подравнени наляво малки символи) Номер на текуща сесия |  |
| QQQQQ | 8 шестнайсетични символа (0 подравнени наляво малки символи) представяне на последователен номер на събитието. Този номер може да се използва като препратка към NVT интерфейса, за да идентифицира доказателствения запис за прехвърляне. |  |
| UUUUU | 8 шестнайсетични символа, времето на заснемане на изображението в UTC (секунди) | 1 |
| mmm | 3 шестнайсетични символа, времето на заснемане на изображението в UTC (мсек) | 1 |
| LLLLLL | 6 шестнайсетични символа, цялата дължина на частта с данните от резюмирания запис |  |
| ididid | осем шестнайсетични символа (променлива дължина) идентификатор на името на файла. Този идентификатор се използва като препратка към записите, запазени на компактна флаш памет. |  |
| \n |  |  |

Полето ididid е необходимо за системи, които прилагат протокол на изтегляне при прехвърляне. Това поле е локална препратка към данните и трябва да се подава с всяко прехвърляне или заявка за изтриване. Това поле може да не се генерира на системи, които изпращат данни към хоста.

За резюмирания запис не се прилагат удостоверяване или кодиране. Процесът, отговорен за прехвърляне на резюмирани записи работи с по-голям приоритет от този, който е отговорен за снимковите записи. Затова в рамките на TCP/IP стека, резюмираните записи може да пристигнат преди съответните снимкови записи.

При наличие на неуспешна комуникация и резюмираните записи не могат да бъдат прехвърлени към хост системата, записите се запазват в локалното непроменливо запаметяващо устройство за последващо препращане към хост системата.

Ако са включени потвържденията, системата ще очаква прехвърлянето на записа да бъде потвърдено. По дадения канал няма да бъдат прехвърляни следващи записи, докато записът не бъде успешно потвърден или докато не бъдат изпълнени всички повторни опити.

## Пакетни резюмирани записи

Някои приложения – по-специално тези, които използват комуникация с ограничен капацитет, като GPRS може да имат нужда от механизъм за прехвърляне на VRN заедно с минималните данни, необходими да се запази определен Доказателствен запис. Процесът на пакетни резюмирани записи предлага тази възможност като използва съкратен формат на Резюмиран запис и като пакетира записите, за да използва ефикасно комуникационния капацитет.

Пакет с резюмирани записи може да включва определен брой превозни средства (max\_batch\_size) или превозните средства в определени времеви рамки (bsr\_interval). Системата ще приготви пакет и ще нареди за прехвърляне пакет от резюмирани записи, когато е достигнат първият от тези лимити.

Камерата ще запази определен брой пакети в опашката (bsr\_nlog), докато бъдат прехвърлени успешно. Ако може да бъде отворена връзка, но прехвърленето на определен пакет е неуспешно при три опита, пакетът се счита за повреден и се изчиства. Ако данните не могат да се изпратят, когато има bsr\_nlog пакета в опашката, най-старият пакет се презаписва.

Когато е включена сигурността, отделни записи в пакета се кодират (AES256 в CBC режим с SR ключ 1 и IV=0) щом създадат. Това гарантира, че само кодирани данни се запазват в камерата. Когато се сглоби пълен пакет, се генерира hmac удостоверителен подпис (като се използва SR ключ 1) и се добавя, а след това целият пакет се кодира повторно (AES256 с SR ключ 2).

Форматът на пакета включва:

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

Това се сравнява с над 400 байта на VRN за пълни Резюмирани записи.

Форматът на хедъра с чист текст е:

mmm SSSS ssss nnnn zzzz cccc ii\n

където

|  |  |
| --- | --- |
| mmmm | магия |
| SSSS | site\_id |
| ssss | номер на сесия |
| nnnn | брой на VRN записите в пакетния блоб |
| zzzz | размер на блоб |
| cccc | crc16 на кодирания блоб |
| ii | нарастваща стойност – прави цикъл на bsr\_nlog – посочва номера на пакета и се използва от процеса на потвърждение/отрицателно потвърждение като номер на събитието |
| \n | нов ред |

Всяко отделно VRN в пакета има следната форма:

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

Обърнете внимание, че пакетът включва съкратени форми на полетата за класификация и статус.

## Потвърждение

Ако системата е конфигурирана да изисква потвърждения на резюмирани и доказателствени записи, те ще бъдат създавани, като се използват данните в хедъра на записите.

Потвърждението е един ред ASCII текст със следната форма:

AAAA,SSSSS,sssss,QQQQQQQQ\n

където:

|  |  |  |
| --- | --- | --- |
| AAAA | Флаг за потвърждение / отрицателно потвърждение. Може да съдържа причина за отрицателното потвърждение. |  |
| SSSSS | Идентификатор на местоположението – променлива дължина от макс. 30 символа – формиран от свързването на id на камерата и id на пътната лента (виж по-горе) |  |
| Sssss | номер на сесията - 8 шестнайсетични символа |  |
| QQQQQ | 8 шестнайсетични символа (0 подравнени наляво малки символи) представяне на последователен номер на събитието. |  |
| \n | край на записа |  |

Дефинираните флагове за потвърждение са:

|  |  |
| --- | --- |
| A000 | записът е потвърден – локалните данни могат да бъдат изчистени |
| A001 | Записът не е приет, моля, изпрати отново. |

Ако не се получи потвърждение за ack\_timeout мсек, системата ще направи опит да изпрати отново данните. Ако половината от ack\_retries опита за повторно прехвърляне са неуспешни, системата ще затвори настоящата връзка, ще отвори отново връзката и ще опита да изпрати отново данните. Ако не успее, хостът ще бъде отбелязан като офлайн. Тогава ще бъде направен опит да се препратят данните към алтернативен хост. Ако са направени ack\_retries опита без успех, данните ще бъдат изчистени или прехвърлени съответно към файловата система.

Ако опитите да се прехвърли снимката не успеят с получаването на съобщенията за отрицателно потвърждение (NAK), след ack\_retries опита записът ще бъде счетен за повреден и ще бъде изчистен.

Ако се изисква потвърждение (ACK), тогава това АСК трябва да бъде получено преди да стане възможно да бъдат прехвърлени други данни.

Внимание: възможно е даден запис да бъде прехвърлен успешно към вътрешната станция повече от веднъж (напр., ако АСК пристигне след като е започнал повторен опит). Отговорност на вътрешната станция е, съответно, да отбелязва и изчиства дублирани записи.

Също така е възможно даден запис да бъде прехвърлен към главния и към алтернативния сървър. Отново, това е отговорност на вътрешната станция да разреши този конфликт.

# NVT канал

В допълнение на стандартния телнет контролен канал е приложен специфичен контролен канал. Както телнет той използва мрежови виртуален терминален интерфейс (NVT). Но за разлика от телнет този канал не преговаря за отразяване, изтриване и др. и не отразява никой вход. Използвания порт може да бъде посочен като конфигурационен параметър. Този канал е приложен за интерфейси машина-машина, спициални интерфейси за вътрешна станция, DSRC или оборудване за контрол на пътна лента. Този интерфейс може да бъде конфигуриран да приема команди или като ASCII стрингове, или като бинарни пакети.

На системи, където хост контролерът може незабавно да установи (за приблизително 5 минути), дали дадено събитие с превозно средство е нарушение, ефикасността на системата може да бъде подобрена като се прехвърлят само нарушенията към хоста. Ако работи в такъв режим, когато възникне снимково събитие, камерата ще информира хост системата посредством резюмирания запис. Снимковите данни ще бъдат запазени на камерата, докато данните не бъдат или презаписани от по-нови данни, или бъдат счетени за остарели. Този канал ще очаква да получи заявки за прехвърляне или изтриване на данни.

Този канал е контролния канал, използван за пренос на заявки към камерата за прехвърляне на записи, когато е включена архитектура на „изтегляне“.

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

Точното приложение на този интерфейс може да зависи от OEM оборудването, към което е прикачен интерфейсът.

## ASCII интерфейс

***5.2***

|  |  |  |
| --- | --- | --- |
| **Команда** | **Описание** | **Отговор** |
| send sss nnnn ididid | Изпраща запис nnnn | * CMD: OK nnnn * CMD:ERR nnnn |
| del sss nnnn ididid | Изтрива запис nnnn | \* CMD: OK nnnn  \* CMD:ERR nnnn |
| init | Изтрива всички записи – инициализира отново системата за съхранение (NYI) | \* CMD: OK nnnn  \* CMD:ERR nnnn |
| hb sss | пулс (NYI) |  |

## Бинарен интерфейс

Този интерфейс е специално за връзки машина/машина. Пълно описание на структурата на данните може да бъде намерено във файла ves\_nvt.h като приложение към този документ. Записите в този текст са само за илюстриране.

Камерата слуша през порт nvt\_port за постъпваща връзка. Веднъж щом се установи връзка, процесът очаква да получи входяща информация поне на всеки nvt\_timeout секунди. Ако не се получи входяща информация за този период, връзката се затваря и процесът слуша за нова връзка. За да поддържа отворена връзка, когато не трябва да се прехвърлят файлове, хостът може да изпраща периодични пулсови съобщения.

Форматът на заявката е:

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;

Хост системата ще даде последователно нарастващ номер на всеки изпратен пакет. ref и ident на записа са информационни полета, дадени като част от резюмирания запис.

Камерата отговаря с:

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;

Пулсовите пакети с във формат:

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;

###### Всички бинарни данни се предават в network byte order

Камерата проверява магическото число и дължината на пакета. Ако някое то тези се окаже без смисъл, вероятно синхронизацията и изключила. Връзката се затваря и камерата очаква нова връзка.

Ако crc не изчислява вярно, се връща резултат за грешка.

Пулсовите пакети се отразяват обратно към хост системата.

Интерфейсът поддържа две главни команди. Тези команди или изтриват определен запис, или препращат записа към хост системата.

##### Кодиране

NVT записите може да бъдат кодирани по желание. Кодирана NVT заявка е просто друг вид NVT заявка, където полезният товар на пакета е кодиран. Когато се разкодира, се получава пакет от стандартен вид заявка.

Ключът за кодирането се създава отново от хост системата при инициирането на всяка NVT връзка. Новият ключ се предава на камерата след успешно преминаване на изпитание и удостоверяване. Ключът се кодира със споделения ключ (KEK)

##### Заявка за изтриване

При получаване на заявка за изтриване процесът ще търси посочения запис първо в ram буферите и, ако не успее, ще опита да осъществи достъп до записа в Компактната флаш памет. Веднъж намерен записът се маркира за изтриване. След това процесът връща ответно съобщение към хоста, което показва, че или записът е намерен и ще бъде изтрит, или че записът не е намерен.

Ако записът е в процес на прехвърляне, записът ще бъде изтрит след успешното завършване на прехвърлянето. В противен случай записът се изтирва незабавно.

##### Заявка за препращане

При получаването на заявка за препращане процесът ще търси посоченият запис първо в в ram буферите и, ако не успее, ще опита да осъществи достъп до записа в Компактната флаш памет. Ако даден запис не може да бъде намерен се изпраща отговор до хост системата, който показва, че записът не може да бъде намерен.

Заявката за прехвърляне на запис по нататък се подразделя. Заявките могат да бъдат наредени за прехвърляне със стандартен приоритет или за прехвърляне с висок приоритет. Веднъж щам даден запис е намерен и преместен в съответната опашка за прехвърляне, се изпраща отговор до хост системата. Той не показва на хост системата, кога прехвърлянето е успешно завършено, а само, че записът е намерен и отбелязан за прехвърляне. По желание съобщение по канала за съобщения ще отбележи успешно прехвърляне.

Ако даден запис успешно е бил прехвърлен от ram буфера, но не е бил изрично отбелязан за изтриване, записът ще бъде автоматично изтрит или при изтичане на давност, или ако ram буферът е необходим за нов запис. Това означава, че записът се запазва в ram за определен период от време, който при нормални обстоятелства може да бъде няколко минути. Следваща заявка за записа ще бъде изпълнена.

Ако не е получена нито заявка за прехвърляне, нито заявка за изтирване, записът ще бъде прехвърлен към Компактната флаш памет при изтичане на давност или, ако ram буферът е необходим за нов запис.

Ако записът е бил успешно прехвърлен от компактния флаш файл, записът ще бъде изтрит незабавно след успешно прехвърляне.

Ако системата работи с протокол на изтегляне с изключени потвърждения за прехвърляне, системата не може да знае, дали даден запис е бил успешно прехвърлен. При тези обстоятелства отговорността за изтриване се прехвърля на вътрешната станция. Вътрешната станция трябва извърши пълно изтриване.

##### Заявка за изчистване

При получаване на заявка за изчистване процесът ще заключи компактната флаш памет и ще започне да трие всички записи, които не са в процес на прехвърляне.

##### Заявка за тест

При получаване на заявка за тест процесът ще създаде нова позиция, за да генерира тестов запис. Този тестов запис ще бъде представен на системата във времето, посочено в заявката.

Ако камерата бъде рестартирана преди да бъде генериран тестовия запис, заявката за тестов запис ще бъде изгубена. Времето на тестовия запис е посочено в UTC секунди/милисекунди и на теория може да бъде по всяко време до 2030 година.

# 6 Сесиен протокол

Виж бележките в приложение Е относно подробности за протокола.

(не се отнася за базовите VES системи)

Системата за разпознаване, в която външната станция с камерата може да работи, не може да действа 24 часа в денонощието. Доказателствените записи са необходими само по време на изпълнителна сесия. Извън сесия камерата може да изпраща резюмирани записи за наблюдаваните превозни средства.

Ключовете за кодиране и удостоверяване на доказателствени записи трябва да се променят често. Затова системата изисква новите сесии да се създават с нови ключове за кодиране и удостоверяване. Сесийният протокол съществува, за да открива и закрива изпълнителните сесии. Този протокол също включва механизми, с които външната станция да удостовери себе си на вътрешната станция и с които вътрешната станция да препрати ключовете за кодиране и удостоверяване на сесията към външната станция.

При стартирането на сесия вътрешната станция изпраща съобщение „стартиране на сесия“ към външната станция. Съобщението за стартиране на сесия включва време на стартиране и дължина на сесията (в минути). Ако времето на стартиране е в миналото, но сесията е все още валидна, сесията ще стартира незабавно. Ако времето на стартиране е в бъдещето, сесията ще е в изчакване и ще стартира, когато достигне времето на стартиране. Само една сесия може да бъде в изчакване.

Ако текущата сесия е активна, когато се получи съобщение за стартиране на нова сесия, текущата сесия ще остане активна, докато сесията се закрие нормално или докато се достигне времето на стартиране на новата сесия.

За да стартира дадена сесия, камерата трябва да е в състояние на готовност да генерира доказателствени записи. Трябва да бъде премината самопроверка на конфигурацията. Не може да бъде зададена индикация за намеса. Часовникът трябва да бъде синхронизиран.

Всяка заявка за сесия, която не отговаря на тези критерии се отхвърля. Камерата ще върне съобщение за отказана сесия, включващо показание за причината за отказ.

Вътрешната станция може да изиска дадена външна станция да закрие текуща сесия по всяко време. Ако прекъснат комуникациите по време на изпълнителна сесия, системата ще продължи да заснема доказателствени записи и, ако това е включено, да запазва записите на компактната флаш памет за по-късно прехвърляне. Ако се напълни паметта, най-старите записи ще бъдат презаписани. Заснемането на доказателствени записи се преустановява по нормален път в края на текущата програмирана сесия.

* 1. Challenge / response протокол

Съобщенията за стартиране/спиране на сесия трябва да бъдат удостоверени. Прехвърлянето на ключ (част от заявката за стартиране на сесия) трябва да бъде удостоверено. Сесийният протокол включва challenge handshake протокол. Вътрешната станция може да изиска удостоверяване на камерата по всяко време. Ако е направена заявка за удостоверяване и е неуспешна по време на изпълнителна сесия, има наличие на намеса и изпълнителната сесия се закрива.

Ако последователността на удостоверяване е неуспешна, поради неверни данни, камерата ще изключи безмълвно. Връзката просто ще се затвори. Никаква информация за причината за изключване няма да бъдат пратени към хост системата.

# 7 Сигурност на системата и цялост на данните

(Не се отнася за базови VES системи)

* 1. Кодиране и удостоверяване

Камерата добавя удостоверителен запис към блоба. Този удостоверителен запис е създаден, като е използван HMAC (*NIST FIPS PUB 198*) стандарт, описан в RFC2104. Откъс от тази препратка е прикачена като приложение – само за информация. HMAC е създаден, като е използван SHA1 (*NIST FIPS PUB 180-1*) и 256 битов ключ.

По желание целият запис на данни (освен хедъра) може да бъде кодиран. Кодиращият цикъл е AES (*NIST FIPS PUB 197*) с CBC и 256 битови ключове.

CBC инициализиращия вектор (IV), използван за прехвърляне на доказателствени записи е зададен изцяло с нули. Това е възможно, тъй като първите 32 байта от доказателствения запис са произволно число. Така IV действително е кодирана версия на първите 16 байта.

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.

Споделеният ключ, използван за удостоверяване на местоположението и за прехвърляне на сесийния ключ, е извлечен от споделена тайна и специфичните данни на местоположението (id на местоположението и сериен номер на камерата). Извличането се извършва, като се използва SHA256 (*NIST FIPS PUB 180-2*) да запази място за 256-битов ключ.

(SHA256 не се използва никъде другаде в интерфейса, тъй като натоварва процесора)

По желание може да се генерира уникална двойка ключове за всеки Доказателствен запис. Когато системата е конфигурирана да прави това, Доказателственият запис е предшестван от ключов блок, съдържащ тези ключове. Ключовият блок е кодиран с ключа за кодиране на сесията.

* 1. Генериране на произволно число

Редица функции за кодиране и удостоверяване изискват употребата на произволни числа. Генерирането на произволни числа (RNG), използвано в камерата се базира на *ANSI X19.17*. Първоначалната посявка се извлича от специфични програмни данни и локалното време. RNG ключът извлича ентропия от следните източници:

* + - броячът на превозни средства
    - сигнал, извлечен от видео шум
    - интервал на tcp/ip пакет

Ентропията и битовете на данните от посявката се смесват с MD5 хеширащ алгоритъм. Системата за кодиране, която се използва за генериране на произволно число е AES256.

Внимание: източниците на ентропия се основават на променливостта, свойствена за поставена камера. Ентропията ще бъде значително намалена в тестова ситуация, в която камера наблюдава статична регистрационна табела и се пуска регулярно. Все пак ентропията се запазва между сесиите и рестартиранията на системата. Виж приложение за статични анализи на RNG продукти.

## Обмяна на ключове

Съществува изискване да се гарантира сигурността за първоначална обмяна на ключове при стартиране на сесия. Ключът за това прехвърляне се извлича от подсигурената споделена тайна.

## Удостоверяване на външната станция

В състояние на покой извън изпълнителна сесия външната станция може да изпраща системни диагностични данни и резюмирани данни за наблюдаваните превозни средства. Те са в чист текст. Все пак преди да може да започне прехвърляне на доказателствени записи, съществува изискване към външната станция да се удостовери на вътрешната станция. Процесът на удостоверяване може да се осъществи по всяко време и при нужда, толкова често, колкото е необходимо, но трябва да се осъществи като част от протокола за стартиране на сесия.

## Споделена тайна

Удостоверяването изисква споделена тайна между вътрешната станция и външната станция. Прехвърлянето на Сесиен ключ изисква споделен ключ. Този ключ се извлича от споделената тайна. Споделената тайна никога не се показва и никога не се прехвърля по мрежата.

При поставяне всяка камера трябва да има качена тази споделена тайна. Тази тайна трябва да специална за id на местоположението и серийния номер на определена камера. Никои две камери или две поставяния на една камера могат да използват една-и-съща тайна. Никоя тайна не може съзнателно да се използва повторно. Генерирането и управлението на споделената тайна е отговорност на вътрешната станция от изключително значение. Разпределението на тази тайна трябва да се извършва ръчно посредством одитиран процес.

Споделената тайна трябва да включва 48 произволно избрани ascii символа, за да осигури 256-битова дълбочина. Предоставя се протокол за прехвърляне на споделената тайна към камерата. Това може да стане единствено посредством серийния порт, който е достъпен единствено,когато камерата е демонтирана. Веднъж качена споделената тайна не съществува в чист текст в камерата и не може да бъде възстановена от камерата. Вътрешната форма е SHA256 хеш на предоставените данни. Виж бележките в приложение Е.

## Цялост на външната станция

Предприети са мерки, за да се гарантира, че се контролира целостта на външната станция и за да се поддържа сигурността на ключовете и целостта на доказателствените записи.

При производството на кода не е наличен механизъм за отпечатване или по друг начин докладване на сесийните ключове от страна на външната станция. Ключовете се съхраняват на локално ниво в криптиран сейф за ключове. Сейфът за ключове се генерира от няколко източника, които са уникални за хардуера на индивидуалната камера (т.е. дори ако сейф за ключове може да бъде копиран от камерата, той няма да бъде четим за друга система).

Ако външната станция засече потенциална намеса, текущата сесия се прекратява незабавно, а сесийните ключове се унищожават. Засичането на намеса може (по желание) да се състои от някое от следните засечени условия:

* + - Неупълномощено отваряне на кутията на пътя
    - Изключване на някой от кабелите на камерата
    - Загуба на мощност – при рестартиране ключовете се унищожават
    - Рестартиране на системата, ръчно или в резултат на грешка на софтуера
    - Връзка на оператора с повече от три грешки при въвеждане на паролата
    - Ако е активиран списък с контрол на достъпа, достъп на TCP/IP, който не е валидиран (опит за хакване)
    - Загуба на пингове на системата Gemini

Ако системата затвори сесия, поради някаква причина, различна от спиране на тока, всички данни, които не са били криптирани до момента, ще бъдат загубени. Това ще бъдат не повече от два или три записа. Въпреки това, всички записи с данни ще бъдат препратени директно или записани на флаш памет, ако не може да се установи връзка със сървърите. При спиране на тока, всички данни, които не са във флаш паметта, се губят.

Ако външната станция засече ситуация, която *може* да компрометира доказателствен запис, тогава се появяват индикации в системите за изключения и пулсови съобщения. Пулсовият статус формира част от всеки доказателствен запис, така че потенциално компрометираните доказателствени записи се маркират индивидуално. Наблюдаваните зони обхващат:

* + - Оператор на линия на telnet порт или уеб страница
    - Оператор наблюдава системата чрез клиент или интерфейси на визьора
    - Всяка промяна в системната конфигурация. При поява на подобна промяна всички следващи записи в тази сесия получават съответната индикация.

## Собствена проверка на системата

За системите на прилагане съществува изискване да гарантират, че настоящата системна конфигурация е правилна и валидна, преди да може да стартира дадена сесия за изпълнение и да бъдат генерирани валидни записи за изпълнение.

В този контекст, “конфигурация” отговаря на набора от файлове, включващ:

#### инсталирано програмно изображение

* + - инсталирано ANPR изображение
    - файл за динамична конфигурация
    - списък с контрол за достъп

Приложението изчислява CRC на активната конфигурация и го сравнява със CRC на копието на конфигурацията на флаш паметта и референцията на CRC в EEPROM. За валидни системи всички CRC трябва да съвпадат. Ако CRC не съвпадат, заявката за стартиране на сесията ще бъде отказана.

Както вече беше посочено в текста, промяната на конфигурацията по време на сесията ще доведе до поставяне на индикация за всяко следващо нарушение и обобщен запис в рамките на тази сесия.

Когато е направена преднамерена промяна на конфигурация, за да бъде валидна системата, архивираното копие на конфигурацията във флаш паметта трябва да бъде актуализирано, както и копието на CRC в EEPROM. Промяната на копието на EEPROM изисква употребата на кода за сигурност на системата.

При стартиране на нова сесия за изпълнение, сесията приема съобщение и връща копие от CRC на системната конфигурация към вътрешната станция.

Може да бъде изискана индивидуална процедура за предоставяне на актуализиран CRC на вътрешната станция за проверка валидността на външна система.

* 1. Списък с контрол на достъпа

Всички връзки на камерата през TCP/IP порт могат да бъдат валидирани спрямо списък с контрол на достъпа. Списъкът с контрол на достъпа се определя с:

>> sys set access\_list filespec

Ако е определен списък с контрол на достъпа, той трябва да е наличен. Това е критичен файл. Този файл се записва и се извиква от флаш паметта, където са извикани функциите key update /обновяване на ключ/ и key restore /възстановяване на ключ. При загуба или изтриване на файла, системното приложение няма да бъде достъпно през TCP/IP връзка. Форматът на файла е:

* 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.

Файлът с контрол на достъпа се прочита при първи опит за реализиране на външна TCP връзка. При всяка промяна на файла с контрол на достъпа трябва да се рестартира системата, за да влезнат в сила промените.

В резултат от всяка грешка при валидиране:

* + 1. се отказва опита за връзка
    2. изписва се изключение
    3. намеса

Функцията за контрол на достъпа се прилага само за входящи TCP връзки. Камерата не излага UDP или ICMP интерфейси , които могат да засегнат системната конфигурация. Не-TCP протоколите, които са изложени, са описани по-долу.

|  |  |  |
| --- | --- | --- |
| **Протокол** | **Порт** | **Функция** |
| UDP | конфигурируем (по подразбиране 7) | echo сървър.  Може да бъде зададен като активиран/деактивиран |
| UDP | 123 | SNTP сървър  Камерата може да изпълнява функцията на SNTP сървър. Тази функция може да бъде активирана/деактивирана. |
| ICMP |  | Пинг отговорът винаги е активиран. |

В камерата липсват защити срещу отказ на сервизни атаки. Това се счита основно за отговорност на мрежата.

## 7.6 Засичане на порт

VES интерфейсът на камерата разполага с няколко порт връзки. Тези портове са ограничени или защитени, както следва:

|  |  |  |
| --- | --- | --- |
| **Порт** | **Описание** | **Заб.** |
| Сериен порт | Този порт се появява на един от задните конектори на камерата. Той се намира на > 4м над земята, така че физически достъпът до него е труден. Този порт предоставя фабрична конфигурация и инженерен интерфейс на софтуера. В производствените системи пиновете към този конектор не са свързани. Кабелът и конектора са контролирани с помощта на други проводници. При изключване на кабела – което се изисква за достъп до този порт – възниква съобщение за намеса. | 1 |
| Telnet TCP/IP порт 23 | За достъп до този порт се изисква инженерна парола. Към този порт може да се прави само по една връзка. Връзката в рамките на кутията не е криптирана. Въпреки това, връзките към кутията са защитени със защитна стена и VPN. Връзката с този порт може да генерира изключение и да постави индикация в статуса на оператора на линия. | 2,3,4 |

|  |  |  |
| --- | --- | --- |
| **Порт** | **Описание** | **Заб.** |
| http TCP/IP порт 80 | За достъп до този порт се изисква инженерна парола. Към този порт могат да бъдат реализирани много връзки едновременно. Връзката в рамките на кутията не е криптирана. Въпреки това, връзките към този порт са криптирани със защитна стена и VPN. Връзката с този порт може да генерира изключение и да постави индикация в статуса на оператора на линия. HTTP услугата може да бъде деактивирана по желание при системи, за които се счита, че тя се свързва с прекалено високо ниво на риск. | 2,3,4 |
| TCP/IP порт 113 | Обратна проверка на email. Вижте RFC1413. Работи, ако са активирани email функциите. |  |
| ICMP пинг | Всяко получено и валидно пинг съобщение ще бъде повторено. |  |
| UDP конфигурируем ехо порт (по подразбиране 7) | Ако е активиран, този порт повтаря всички получени съобщения. По подразбиране, тази функция е активирана. |  |
| TCP/IP 3570 | Този порт осигурява средство за контрол на връзката между клиентското приложение PIPS, показващо изображения, и засечените регистрационни табели. (Входящите данни по този порт могат да изменят конфигурацията на камерата. Ако е направена връзка през този порт, в статуса на оператора на линия се поставя индикация.) |  |
| TCP/IP порт 3577 | Връзка към този порт ще получи стрийминг от данни, които се състоят от оперативни, диагностични съобщения и съобщения за грешка. Входящите данни към този порт се игнорират. |  |
|  |  |  |
| TCP/IP порт визьор (порт 9000) | Този порт предоставя стрийминг от изображения за диагностика и наблюдение. Входящият стрийминг от данни през този порт може да извърши ограничителни промени на системнаа конфигурация (избор на изображение, компресия и т.н.). Връзката с този порт може да генерира изключение и да постави индикация в статуса на оператора на линия. | 4 |
| TCP/IP порт моментна снимка (порт 9001) | При връзка с този порт се получава едно единствено изображение от камерата. Входящите данни се игнорират. |  |
|  |  |  |
| Ключ на приложението порт  (по подразбиране 10003) | Този порт е контролният порт на сесията. Той работи с частен протокол. За всяка операция, проведена през този порт, е необходима проверка. |  |
| Стартиране на приложение  порт  (по подразбиране 10006) | Ако е активиран, този порт приема съобщения за стартиране от устройство за контрол на съответната лента или DSRC оборудване. Той работи с частен протокол. Всички съобщения, които не отговарят на частния протокол, се игнорират. Конфигурацията на системата не може да бъде променена през този порт. |  |
| Приложение NTV порт  (по подразбиране 10002) | Този порт предоставя приложение за специален контрол, при който се получава ограничаващ комплект от команди от локално свързаното устройство за контрол на лентата или от оборудване на вътрешната станция. |  |
| приложение порт за съобщения  (по подразбиране 10004) | Връзка към този порт получава напредъка на приложението и информационните съобщения. Входящите данни се игнорират. |  |

Заб.:

1. При тестови системи този порт ще бъде свързан обратно към кутията на пътя.
2. При опит за премахване на Етернет връзката от камерата, за да се създаде директна връзка към камерата ще се появи предупреждение за намеса.
3. Препратките към „кутия“ се отнасят за заграждението на пътя, в което се помещава комуникационната инфраструктура. Обикновено това е влакнеста или медна връзка, наета DSL или ADSL линия. Връзката към комуникационната инфраструктура обикновено се извършва чрез защитна стена/рутер и предоставя допълнителна сигурност под формата на VPN.
4. Мрежовата инфраструктура и кутията на пътя се предоставят от друг, поради което не са обект на настоящия документ.
5. Ако системата комуникира чрез GPS или друга безжична услуга, то оборудването на пътя може да се състои единствено от захранване.
6. Косвено се допуска, че ако съществува, кутията на пътя се охранява отделно и съществува механизъм, който да индикира, че е направен опит за намеса в камерата.
7. Получава се системно изключение или на всяка връзка, или на връзки, направени по време на сесията за изпълнение.
8. Всички TCP/IP портове за слушане могат да валидират входящу връзки с помощта на списъка за контрол на достъпа.

1.7 Пароли

VES системата използва няколко пароли, за които стана дума по-горе в текста.

##### Главна CLI парола

Системата може да бъде конфигурирана да изисква парола преди да бъде направен какъвто и да е достъп до нея през CLI интерфейса. Ако бъде настроена така, хеш на тази парола се съхранява в EEPROM. Тази парола трябва да бъде въведена за достъп до системата, както в нейното приложение, така и в режим EEPROM. Въведената парола се вкарва през еднопосочна хеш функция (sha1), преди да бъде сравнена със съхранените данни. Хеширането на паролата гарантира, че дори да бъде получен достъп до съдържанието на паметта, паролата няма да може да бъде пресъздадено,при преглед на съдържанието на RAM паметта.

Тази парола е с основно значение при поддържане сигурността на системата. Ако тази парола бъде загубена, камерата може да бъде възстановена единствено чрез процедура, извършвана през серийния порт. За да се поддържа целостта тази процедура ще доведе до тотална загуба на всички ключове (включително ключ за споделена тайна и криптиращ ключ), всички данни за конфигурацията и всички програмни изображения на приложението.

Тази парола може да има различна дължина и може да се състои изцяло от печатни значи. Отговорност на оператора е да издава и налага паролата.

##### Парола за уеб страница

Съществува вероятност в всички осигурени приложения да бъдат деактивирани HTML интерфейса и системите за управление на уеб страница. Все пак, ако операторът реши да запази уеб интерфейса, той може да зададе парола, която трябва да бъде въвеждане за получаване на достъп до интерфейса.

## 7.7 Коментари за съответствие с правилника на Полицейското управление за научно развитие (PSDB)

Оригиналният правилник на PSDB (PSDB 3/96) е изготвен преди девет години. Технологията се развива и процедурите, заложени в този правилник, вече не са приложими.

В оригиналния правилник е зададено криптиране с употребата на 56-битов DES с 64-битов CBC. Индустрията вече не счита DES алгоритъма за предоставящ адекватна сигурност – в средата на деветдесетте години в рамките на 3 часа можеше да бъде създадена специална машина за разбиване на шифъра. Направените актуализации през 2002 година изказват предположение за използване на троен DES. Системата предлага използването на по-силния AES шифър (*NIST FIPS PUB 197 version ???*) със СВС верига и 256 битови ключове.

В ръководството е заложена употреба на DES-MAC, като код за удостоверяване. Този интерфейс предлага употреба на текущия промишлен стандарт, който е базиран на HMAC (*NIST FIPS PUB 198*) и използва SHA1 (*NIST FIPS PUB 180-1*). Използва се отново 256 битов ключ.

Използват се отделни ключове за удостоверяване и криптиране, въпреки че, по желание може да се използва и един ключ за двете операции. Но това не е препоръчително.

По желание могат да бъдат създадени уникални ключове и да бъдат използвани за удостоверяване и криптиране на тялото на всеки доказателствен запис. Ако е активирана тази услуга, тогава всяка уникална двойка ключове се криптира с ключа за криптиране на сесията и се добавя към всеки доказателствен запис.

По желание могат да бъдат криптирани контролните NTV съобщения. Ако е активирана тази функция, при първоначалното отваряне на NTV канала ще се изисква последователен CHAP, преди ключът да премине през линка.

В ръководството е зададена употребата на прехвърляне на ключове с помощта на тройно криптиран DES и трети и четвърти ключ, КЕК (ключовете за криптиране на ключове), които трябва да се разпространят ръчно до обектите на външната станция с помощта на контролиран процес, записите от който формират част от веригата с доказателства.

Този интерфейс гарантира получаването на 256 битови КЕК от споделената тайна, използвана за удостоверяване. Сесийните ключове се прехвърлят с помощта на този 256 битов КЕК. Съществува защитена процедура за ръчно инсталиране на споделената тайна от конкретен хардуерен порт и само от този порт. Отговорност на потребителя е споделената тайна да разполага с разумна битова дълбочина. Но дори и да бъде заредена тайна с ограничена битова дълбочина, тези входящи данни се комбинират с (позната) селекция и се превръщат в 256 битов обект с помощта на SHA256 (*NIST FIPS PUB 180-2*) , за да се гарантира, че този ключ притежава адекватна битова дълбочина.

Съществува изискване за гарантиране на това към всеки доказателствен записа да се прилагат проверка и криптиране, възможно най-скоро след заснемане. Създаването на проверен и криптиран доказателствен запис има голям приоритет в камерата/системата за автоматично разпознаване на регистрационни табели. При нормални обстоятелства доказателственият запис се създава в рамките на три секудни след събитието на заснемане. При пиково натоварване това може да отнеме повече време, но никога не трябва да надвишава десет секунди. Може да се използва тестов софтуер, който да покаже хистограма на латентността между заснемането и криптирането.

Актуализираното ръководство на PSDB твърди, че контролните интерфейси трябва да бъдат криптирани. По желание, NTV каналът може да бъде криптиран (вижте по-горе), за да изпълни това изискване. Дистанционните интерфейси към камерата/ системата за автоматично разпознаване на регистрационни табели не се криптират в рамките на системата. Очаква се тези интерфейси да се обслужват от кутията по сигурността на външната станция чрез VPN, което напълно отговаря на изискванията на PSDB. При засичане на неоторизиран достъп до комуникационната кутия или кабелите на камерата, незабавно се прекратява текущата сесия и се изтриват текущите сесийни ключове. В резултат на всички засечени опити за неоторизиран достъп до комуникациите на камерата през мрежата (опити за хакване) се появява изключение. VES ръководството приема, че камерата използва криптиране единствено за прехвърляне на доказателства при активна сесия. Извън сесията не се създават доказателствени записи.

Въпреки това, все пак съществува възможността за това *оторизиран* достъп да компрометира системата. Съответно, при всяка връзка към контролния интерфейс на потенциалната камера/ системата за автоматично разпознаване на регистрационни табели в системата се получава изключение. Докато контролната връзка е активна, всички доказателствени записи ще получават маркер, който ще индикира, че може да са компрометирани. Всички промени в системната конфигурация, направени по време на активна сесия, ще получат индикации, която ще се прилага при всеки следващ запис в рамките на тази сесия. По този начин вътрешната станция и бек офиса могат да поемат отговорност да категоризират повторно всички записи, за които се счита, че може да са компрометирани.

# Възможност за самовъзстановяване – засичане на грешки в хоста & комуникациите

Могат да бъдат посочени два първични IP адреса. VES интерфейсният софтуер създава хоста като го пингва. Хост, който не отговори на пинга, се счита не съществува. Пингът представлява ICMP или UDP заявка за копиране (ехо заявка). UDP ехо порта по подразбиране е порт 7, но може да бъде специално зададен. По този начин хост системите могат да упражняват контрол помежду си, при който да приемат данни от камерата. Пинг контролите са:

|  |  |  |
| --- | --- | --- |
| sys set ping mode | n | Където n=0 за ICMP и n=1 за UDP |
| sys set ping port | nnnn | Зададен UDP порт за използване от UDP ехо заявките equests |
| ves set ping interval | nnn | секунди между опитите за пингване за проверка дали са на линия хост адресите |
| ves set ping wait | nnn | максималният брой ms, които се изчакват за пинг отговор, преди да се реши, че пингът няма отговор |
| ping aaa.bbb.ccc.ddd |  | CLI тест за изпращане на ICMP пинг към конкретен хост |
| ves ping  aaa.bbb.ccc.ddd |  | CLI за изпращане на пинг към конкретен хост чрез конфигуриран пинг режим и порт |

Заб.: Когато VES интерфейсът пингне даден хост, се генерират две пинг събития, тъй като пинг през ICMP или UDP протокол се счита за ненадежден протокол.

Ако бъде избран UDP пинг (UDP ехо заявка), тогава UDP пакетът се състои от полезни данни в полезния товар. Полезният товар представлява ASCII стринг под формата на

aaa.bbb.ccc.ddd aaa.bbb.ccc.ddd sssss eeeee nnnnn SSS BBB V site\_id ssssss xxxxx ii

където

|  |  |  |
| --- | --- | --- |
| **Поле** | **Описание** | **Заб.** |
| aaa.bbb.ccc.ddd | IP адрес на целта |  |
| aaa.bbb.ccc.ddd | IP адрес на хоста | 3 |
| Ssssss | UTC секунди (шестандесетични) |  |
| Eeeeee | Изтекли местни тикове (ms) шестандесетични |  |
| Nnnnn | Брояч на събитията в пинг системата (шестандесетични) |  |
| SSSS | Сериен номер на устройството (десетичен) |  |
| BBB | Софтуерен номер (десетичен) |  |
| V | Символ, посочващ пинг източникът | 2 |
|  |  |  |
| site\_id | Обектен идентификационен номер | 1 |
| Sssssss | Статус на локалното приложение | 1 |
| Xxxxxxx | Разширен локален статус | 1 |
| Ii | Брояч на идентификациите на пинг приложението десетичен – в рамките на 99 |  |
|  |  |  |

1. Приложението може да генерира други пингове, преди да отвори връзката. Тези полета може да не са налични. Вижте забележка 2.
2. . Съществуват много на брой възможни източници на заявки за копиране на съобщенията в рамките на VES приложението и на други приложения на камерата. Данните, предхождащи това писмо, се генерират с помощта на модул за ехо заявки и ще бъдат еднакви за всички ехо заявки. Данните, следващи тази индикация, са специфични за приложението. VES интерфейсът генерира два вида ехо заявки. Видът V се генерира периодично, за да създаде състоянието на хост системите. Видът О се генерира преди всеки опит за връзка. Видът О изисква да не се съдържат допълнителни данни.

3). Източникът на UDP ехо заявка, разбира се, може да бъде изтеглен от приемника. Въпреки това, може да са налице NAT между камерата и хост системата. При това условие, данните позволяват на ехо сървъра да създаде както явен, така и действителен адрес на източника.

4)

Ако текущият хост излезе от линия (не успее да отговори на две последователни пинг заявки), всички последващи транзакции към този хост ще се опитат да завършат. След това връзката ще бъде затворена. Системата ще опира да отвори нови връзки, както се изисква от другия хост, ако той е на линия. По този начин, една промяна на вътрешната станция би деактивирала ехо сървъра на една вътрешна станция, след това би държала вътрешната станция активна за даден период, през който да се разреши завършването на чакащите събития.

При промяна на статуса на хоста, всички отворени връзки на VES ще бъдат затворени и отворени отново.

Портът за съобщение и NTV порта изслушват. Все пак, тези портове също ще бъдат затворени при гореописаните обстоятелства, като по този начин позволяват на съответната вътрешна станция да създаде нови връзки.

Ако основният хост е на линия и приема данни, когато алтернативният хост дойде на линия, не се предприема никакво действие.

Ако алтернативния хост е на линия и приема данни, когато основният хост дойде на линия, тогава след приключване на всички текущи транзакции по всички портове, връзката се затваря и се отваря отново през основния порт.

При промяна на статуса на хоста се генерира съобщение за изключение.

Заб.: всеки порт на камерата (освен уеб интерфейсът) приема само по една входяща връзка.

Два хоста могат да бъдат посочени за синхронизиране на времето, независимо един от друг. Системата винаги ще опита да синхронизира основният, преди второстепенният.

Ако даден опит за изпращане на данни към хоста остане неуспешен (както е описано в този документ), тогава този хост се маркира като извън линия и ако алтернативният хост е посочен като достъп, ще се извърши опит за свързване с него. Пояснение – ще се направи втори опит, ако хостът все още е на линия.

SNTP се държи по различен начин. SNTP процесът винаги опитва основния преди второстепенния времеви сървър.

Възниква проблем при синхронизацията, единствено когато нито един от сървърите не отговаря. Проблем в синхронизацията се докладва, само когато с никой от сървърите не може да бъде установена връзка за период, надвишаващ двадесет минути.

# Канал за съобщения

#### Интерфейсът разполага с канал за съобщения, който предоставя данни за напредъка и съобщения за статуса на операциите за прехвърляне на записи. Този канал за съобщения се появява на отделен конфигурируем TCP/IP порт. Ако портът не е свързан, това няма да се отрази на поведението на системата. Съобщенията се форматират по строга схема, така че да улеснят машинното разчитане. Съобщенията се предават или в обикновен текст или в криптирано писмо.

## Криптирано писмо

Съдържанието на криптираните съобщения е идентично на съобщенията в обикновено текстов формат, описани по-долу, но всяко съобщение е поставено в криптирано писмо.

Когато дадена връзка в канала за съобщения бъде отворена за първи път, камерата е предизвикана от хост системата (CHAP протокол). Ако отговорът на предизвикателството задоволи хоста, че комуникира с очакваната камера, тогава с помощта на КЕК, получен от споделената тайна, се предава уникален и краткотраен ключ за връзка. Този ключ за връзка се използва за криптиране на всички следващи съобщения, предавани по времето на текущата връзка. Съобщенията се криптират с помощта на AES256.

Каналът за съобщения ще бъде затворен от камерата автоматично а) ако връзката не е използвана в рамките на 60 секунди (може да се конфигурира изтичане на съобщенията msg\_timeout) или b) ако връзката е била активна в период, надвишаващ 60 минути. Така се гарантира, че хост системата е принуждавана периодично да създава нова връзка със свързан нов краткотраен ключ.

Хост системата също изпълнява политика за затваряне и повторно отваряне на връзка, с цел иницииране на нов краткотраен ключ.

Форматът на писмото, в което се съдържа съобщението, е следният:

#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;

#### където **msg** е съобщението в обикновено текстов формат, описано по-долу. Полетата **nonce** и **msg** са криптирани чрез AES256.

## Обикновен текст

Съобщенията ще:

бъдат ASCII текст

завършват с нов ред

започват със специален идентификатор със следния формат:

Lnnn

където L е главна буква, една от следните:

напр.

M напредък на съобщението

W предупредително съобщение

E съобщение за грешка

P съобщение за паника

#### nnn е уникален цифров идентификатор на съобщението

P123 PANIC @ 0x123abcd widget failed

Системата е разширяема. На практика, единствените съобщения, които към момента са изпълними за това приложение, са:

|  |  |  |
| --- | --- | --- |
| **Съобщение** | **Описание** | **Заб.:** |
| M000 n keep alive | Продължава да изпраща реални съобщения на всеки *timeout* секунди, където *timeout* е конфигурируем параметър, описан по-горе. n е десетично нарастване на всяка трансмисия | 1 |
| M002 site ssss eeee | Остаряло съобщение  site – обектна идентификация  ssss – номер на сесията  eeee – номер на локалното събитие |  |
| M004 c iiiii uuuuuuuu | Посочва, че е зададеният TCP/IP пренос на изображение за камера c идентификатор iiiiii при времеви печат uuuuuuuu (шестнадесетичното UTC време, свързано с изображението) е завършен |  |
| M005 | Стартирала е нова сесия |  |
| M006 | Сесията е завършила нормално |  |
| M007 | Сесията не е завършила нормално |  |
| M008 tbd | Включен е текущият сървър | 2 |
| M014 c uuuuuu sss nnnnn ppppp ididid | Показва, че е стартирано събитие в камера c във време uuuu секунди, sss ms с поредност nnnn и регистрационен номер pppp |  |
| E001 nnnn string | Игнорирани данни - Запълнена опашка от изображения за обработване |  |
| E002 nnnn string | Игнорирани данни - Неуспешно свързване (пингване) със сървъра |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Съобщение** | | | **Описание** | **Заб.** |
| E003 nnnn string | | | Игнорирани данни - Сървърът е видим (пингва) но неуспешно свързване с трансферното гнездо |  |
| W001 tp | vvvvv | string | Данните са прехвърлени в буферния диск - Запълнена опашка от изображения за обработване |  |
| W002 tp | vvvvv | string | Данните са прехвърлени в буферния диск - Неуспешно свързване (пингване) със сървъра |  |
| W003 tp | vvvvv | string | Данните са прехвърлени в буферния диск - Сървърът е видим (пингва) но неуспешно свързване с трансферното гнездо |  |
| W010 tp | rrr | | Получавани са задействащи сигнали със скорост, която надвишава системната спецификация. Има вероятност събитията да не се обработват правилно. |  |
| H001 tp | sssss | bbb f | Копие на пулсови данни – вижте по-долу за обяснение.  Заб.: Трябва да сте активирате функцията за пулс, за да бъде създадено това съобщение. |  |
|  | | |  |  |

Заб.:

* + 1. Това съобщение може да се използва за наблюдаване на системата с цел получаване на потвърждение, че лентата е напълно функционална.
    2. Индикация, че е включен основният сървър.

# Пулсов канал

* 1. Формат на пулсовото съобщение

Ако опцията *heart\_beat* е зададена с ненулева стойност, тогава на всяко количество *heart\_beat* пулсови секунди, 372 ще отваря връзка към определената хост система и ще изпраща пулсови съобщения. Връзката се затваря след прехврълянето. Със самото си съществуване пулсовото съобщение показва, че камерата работи и че мрежовата връзка е наред. Съдържанието на пулсовото съобщение дава кратка индикация на оперативния статус на камерата. Полето на пулсовия статус също е достъпно през основното изображение на доказателствения запис и през обобщения запис. Битовите полето на това поле на статуса показват всички потенциални слабости в системната сигурност, които биха могли да компрометират целостта на доказателствения запис (напр. отворена врата на кутията, инженер, свързан през telnet, несинхронизирано време).

iiii,ssssssss,bbb,f\n

където:

|  |  |
| --- | --- |
| iiii | Идентификатор на станцията – това е обектната идентификация на 372 |
| ssssssss | Статус на станцията – шестнадесетично число . 0 е OK, всяка ненулева стойност може да индикира потенциален проблем на системата. Точният формат е описан по-долу. |
| bbb | Пулсов брой – десетично число. В рамките на 999. Получаващата система може да провери поредността, за да засече наличието на евентуални проблеми по мрежата или в системата. |
| f | Десетично число за грешка при изпращане на пулс. При успешно изпращане, това поле се нулира. Ако стойността постоянно е нула, това може да е индикация за постоянен проблем в мрежата. |

напр.:

1234\_1,00000000,876,0

което посочва

* + - станция 1234\_1
    - статусът на станцията е 0, което е OK
    - изпратени са 876 пулсови съобщения
    - липсват грешки при изпращане на пулсови съобщение от последното успешно изпращане

Ако променливата not\_alive има ненулева стойност, когато броят на последователни неуспешни опити да се изпратят пулсови съобщения достигне това число, тогава системата се опитва да се възстанови. Най-възможният изход е рестартиране на системата.

## 10.1 Доклад на статуса

Налични са два статуса, представляващи 64 бита статус.

Обърнете внимание, че поредицата от битове може да не посочва истинска стойност. Могат да бъдат зададени един или няколко от следните битове, за да посочват вътрешния системен статус:

|  |  |  |
| --- | --- | --- |
| **Битово поле** | **Описание** | **Заб.:** |
| 0x00000001 | последно неуспешно изпращане на пулсово съобщение |  |
| 0x00000002 | Основният хост е извън линия |  |
| 0x00000004 | Алтернативният хост е извън линия |  |
| 0x00000008 | при последния трансфер на изображение е докладвана грешка |  |
|  |  |  |
| 0x00000010 | спиране на засичането на превозни средства за камера 1 | 1,2 |
| 0x00000020 | Вратата на кутията е отворена – НЕОТОРИЗИРАН ДОСТЪП |  |
| 0x00000040 | Обобщаващият линк към хоста е затворен | 1 |
| 0x00000080 | Линкът за изображения към хоста е затворен | 1 |
|  |  |  |
| 0x00000100 | Системата е направила топъл рестарт. Това е първото успешно изпращане на пулсово съобщение след статирането. Този бит може да бъде изчистен при следващите съобщения. |  |
| 0x00000200 | Няма активна сесия за доказателствени записи |  |
| 0x00000400 | В рамките на текущата сесия са направени промени в конфигурацията |  |
| 0x00000800 | Оператор е на линия. |  |
|  |  |  |
| 0x00001000 | Камера 1 докладва грешно състояние | 3 |
| 0x00002000 | Не се използва |  |
| 0x00004000 | Не се използва |  |
| 0x00008000 | Не се използва |  |
|  |  |  |
| 0x00010000 | Системата все още не е направила опит да фиксира времето през SNTP |  |
| 0x00020000 | SNTP сървърът не отговаря |  |
| 0x00040000 | Системата не е успяла да синхронизира с SNTP сървъра |  |
| 0x00080000 | Системата не е успяла да фиксира времето след 20 опита за синхронизация | 4 |
|  |  |  |
| 0x00100000 | Данните са препълнили вътрешната памет или сървърът не може да бъде достигнат. Изображенията се прехвърлят на локалния диск. |  |
| 0x00200000 | Във вътрешната памет се съхраняват изображения за прехвърляне към хоста |  |
| 0x00400000 | Вътрешната система на диска е препълнена. Доказателствени записи са били презаписани. |  |
|  | . |  |
| 0x01000000 | Дискът не е готов |  |
| 0x02000000 | Дискът не съществува или не отговаря на заявки |  |
| 0x04000000 | Работата на диска е върнала грешка |  |
| 0x08000000 | ПРАЗНО |  |
| 0x10000000 | ПРАЗНО |  |
| 0x20000000 |  |  |
| 0x40000000 | Системния и референтен CRC не си съответстват при рестарт на системата |  |
| 0x80000000 | Системата е направила студен рестарт в резултат на спиране на тока или при намеса на оператор. Този бит ще бъде изчистен при последващи съобщения. |  |

Забележки

1. При слаб трафик е нормално да се появяват тези съобщения.
2. Този бит се настройва, когато във входящия канал не е поставено изображение в рамките на период, надвишаващ timeout секундите.
3. Този статус се настройва, ако входът на камерата посочва, че няма сигнал за синхронизиране.
4. Това може да е нормално за кратък период от време, докато ако системата е била без синхронизирано време за период, надвишаващ няколко дни, както поради прекъсване на връзката, така и поради изключване. Фиксирането на времето трябва да бъде извършено в рамките на два цикъла на системата след стартиране. Ако това не се случи, тогава съществува реална грешка.

Със следващата спецификация се дефинира разширен статус от 32 бита.

|  |  |  |
| --- | --- | --- |
| Битово поле | **Описание** | **Заб.** |
| 0x0000000f | Запазен - подсистема за GPS синхронизиране на времето |  |
| 0x10000000 | Статус за синхронизиране на времето на сесията | 1 |
| 0x20000000 | Статус за оператора на сесията | 1 |
| 0x40000000 | Статус за конфигурацията на сесията | 1 |
| 0x80000000 | Статус за намеса в сесията | 1 |

Заб.:

* 1. Тези битове се изчистват след края на сесията и се задават в рамките на сесията, ако са настроени някои от съответните аларми. Изключението в края на сесията ще съдържа копие от тези битове, преди да бъде изчистен статуса. По този начин. краят на изключението на сесията може да се използва за индикация, в случай че по време на сесията са били задействани аларми.

# Канал за задействане

Камерата може да се самозадейства, може да бъде задействана от външен задействащ механизъм, ако има такъв, или да бъде конфигурирана да приеме съобщения за задействане през TCP/IP връзка или връзка през серийния порт.

3М не препоръчва употребата на Етернет задействащи механизми, освен при условие, че трафикът е много бавен.

Камерата слуша за единична връзка през TCP/IP порта. TCP/IP съобщенията са обект на TCP/IP времето за изчакване и може да не са подходящ избор за всички ситуации. При възможност PIPS препоръчва използването на серийни съобщения.

Конфигурацията на серийния канал е трижична, 19200bps, 8 данни, 1 стоп, без паритет и без контрол на потока. Ако бъде свързан към серийния порт на компютър, не се изисква връзка с модем.

Съобщението за задействане се очаква като ASCII стринг в следния формат:

STX n .... \n

където:

|  |  |
| --- | --- |
| STX | Символ за стартиране на съобщението |
| n | Канал за задействане (0-3) (изискван за p357 изпълнения) |
| .... | Зарежда полезния товар на съобщенията (не трябва да съдържа STX или \n и ще бъде игнориран, освен в определени OEM системи) |
|  |  |
| \n | Завършек на съобщението |

Цялата дължината на съобщението за задействане не трябва да надвишава 64 символа. Данните, получавани като полезен товар на съобщението за задействане, се прехвърлят в хост системата и в обобщените записи, и в записите на изображения. При нормални условия, съдържанието на полезния товар на задействащия механизъм се игнорира. Специални OEM системи може да разполагат с код за преобразуване на тези данни.

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.

# 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.

## 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 |

## 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.

# 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.

## 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 |  |

## 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 |  |

## 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.

## 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)* | *Maximu m 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 |

## 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)* | *Maximu m 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. |

* 1. ***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)* | *Maximu m 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). |

##### 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 |

##### amera 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) |

* 1. ***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)* | *Maximu m 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 (%%). |

##### 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). |

##### 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

##### 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)* | *Maximu*  *m 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)* | *Maximu m 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.

# References

## 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 |  |

## 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.

## 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 |

# Appendices

## 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

## 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

* + 1. **ame** ves.exe ves\_session.exe

ves\_trig.exe

ves\_monitor.exe

* + 1. **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 |

##### escription

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.

##### 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.

##### 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 >00010203040506070000000000000000000000000000000000 0000000000000<: 4213441

ves\_image: decryption: 3mS

ves\_image: authenticating with 256 bits of key >000102030405060700000000000000 000000000000000000000000000000000<

ves\_image: authentication OK in 1mS

|  |  |
| --- | --- |
| patch size: | 3341 |
| full size: | 4891 |
| oview size: | 8178 |

ves\_msg: M002 lane\_1 00000004 0000000b

## 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)

## 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.)

1. 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.

1. 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.)

1. 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.

1. 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.)

1. 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.

## 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;

##### erivation 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:

* + - 1. generating an SHA256 hash of the internal shared secret
      2. generating an SHA256 hash of the site\_id and serial number
      3. 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);

##### 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  #define | VES\_MAGIC\_DE  VES\_MAGIC\_KEYS\_B | "TD04"  "TKB4" |  | // diag/excep  // 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  #define | VES\_TAG\_CAMERA VES\_TAG\_LANE | 0x02 | 0x01 |
| #define  #define  #define  // etc | VES\_TAG\_ORIENTATION  VES\_TAG\_CRCS  VES\_TAG\_SESSION | 0x03  0x04 | 0x05 |
| #define  #define  #define | VES\_TAG\_LENGTH\_MASK VES\_TAG\_TYPE\_MASK VES\_TAG\_TYPE\_SHIFT | 0x07  0xf8 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

##### 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.

##### 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

>>

##### 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);

## 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.

##### 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.

##### 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.

## Appendix G - Readability of License Plates

##### 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).

##### 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.

##### 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.

##### equired 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.

##### 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.

120

100

**Performance degradation**

80

60

40

20

0

0 10 20 30 40

**Tilt degree**

#### 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.

# VES ADDENDUM - Radar Interface

## 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.

## 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 |

## 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.

1. radar sensitivity may need to be adjusted from default based on camera range and expected vehicle size.
2. 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.
3. 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.

## 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.

## 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.

## 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)

## 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. |

## 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.

## Error minimisation & correction

##### osine 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 |

* + 1. **orrection 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.

##### 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.

* + 1. **alibration / 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.

## 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.

##### 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

##### 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.

##### 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.

##### 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.

##### 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.

##### 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.