pc works

2013

Operating Systems	Price
Windows XP Professional OEM	7,750
Windows 7 Home Basic OEM	5,000
Windows 7 Home Premium OEM	6,500
Windows 7 Professional OEM (32bit)	7,50
Windows 7 Professional OEM (64bit)	8,250
Windows 7 Ultimate OEM	10,000
Windows 7 Home Premium FULL PACK	7,250
Windows 7 Ultimate FULL PACK	12,500

Office Suites	Price
Office 2011 Home & Student (Word, Excel ,Powerpoint, One Note)	3,950
Office 2011 Home & Business (Word, Excel ,Powerpoint, Outlook, One Note)	10,000
Office 2011 Home & Business for Mac	10,000
Office 2011 Professional (Word, Excel, Powerpoint, Outlook, Access, Publisher, One Note)	25,500

Server Products	Price
Windows SBS 2008 STD 32bit/64bit (5 User) OEM CD	45,500
Windows SBS 2008 STD R2 64bit (5 User) OEM CD	48,000
Windows SBS 2008 Premium (5 User)(Win2003, SQL, Ex, ISA) OEM CD	40,000
Windows SBS 2008 Premium (5 User)(Win2003, SQL, Ex, ISA) FULL PACK	36,200
Windows SBS 2008 Additional CAL (5 user) License	18,500
Windows 2008 Server Additional CAL License MLP	3,650

Antivirus Products (1 year Subscription pack)	Price
Kaspersky Internet Security 2013 - 1 User	750
Kaspersky Internet Security 2013 - 3 User	1,800
Kaspersky Small Office Security Antivirus (1 Servers + 5 Workstation)	6,500
Kaspersky Small Office Security Antivirus (1 Servers + 10 Workstation)	8,000
Quick Heal Antivirus Server - 1 User	2,000
Antivirus Installation & 1 year Support charges included	

Office-Home Accounting Products	Price
Tally 9 ERP Silver (Single)	18,000
Tally 9 ERP Gold (Multiuser)	54,000
Tally 7.2 Silver to Tally 9 Silver Upgrade	7,200
Tally 7.2 Gold to Tally 9 Gold Upgrade	21,600
Tally. NET Subscription Single User	3,600
Tally. NET Subscription Multi User	10,800

Backup Products	Price
Genie Backup Manager Professional 8.0 (Single)	3,500
Genie Backup Manager Professional 8.0 - 5 User Paper License	15,000
Genie Backup Manager Server 8.0	10,500

Security Software	Price
Spook Data Security & Encryption Software (Multiuser)	1,50,000.00

Prices may vary due to exchange rates. All prices inclusive of taxes & delivery .

for Data Security

.....integrated solution for data security

Security....

Spook is an integrated solution for data security. It is a data vault, a data transporter, and a concealment engine all packed into one application. What sets aside it from other applications is its concealment feature.

We are more concerned about your data security than the application features itself; hence Spook uses a unique kind of algorithm to encrypt your sensitive data that can only be decrypted by the correct password entered at the time of encryption. You can choose your password from virtually infinite combinations of about $224^{20} + 224^{19} + 224^{18} + 224^{17} + 224^{16} +$, and so on. In other words, the intruder will have to read your mind in order to unlock the data. What's more?

The data encrypted by Spook can only be unlocked by the correct password no matter what technique is used to decrypt it. No other software, no other human, no other Bot in this world, not even Spook itself can dare unlock it without the correct password. On top of that, the encrypted data looks like a garbage dump while in its encrypted form, and it is technically impossible to know the source of it. In other words, even if the intruder happens to come across the concealed data, he/she would not even know that this has been created by Spook rendering it impossible (even technically) that the data has been encrypted.

FEATURES OF S P O O K

No need to install the software

Protect crucial data with up to 20-character case-sensitive password

Remove trace of your source data beyond recovery

Remove trace of even already deleted data

Compress and archive with encryption

Transfer crucial data in a safe and secure manner

Single-click encryption, archival and sanitization of your important data

SPOOK™ ...integrated solution for data security, backup and transfer

HOW SECURE IS SPOOK???

Some may take our word for it but many users want to know more about the internal workings of the product. This is justified, after all security of their data is at stake. We present this paper to answer the most frequent queries we have received related to the issue. In case the user still has more questions we would be glad to answer them. The paper contains mathematical notations and encryption related terms, the use of which could not be avoided due to the technical nature of the issue covered.

THE ALGORITHM

The algorithm is a process by which data is converted to encoded form. This is primarily done to protect its integrity and avoid malicious use. The algorithm in SPOOK has been specially designed for it, keeping in mind the management and storage of data on computers. It is amazingly fast and can encrypt up to 100 Megabytes per minute (disk access time not included). However the key-points of the algorithm are **data-dependent and password-dependent**. We explain these terms in more details below.

THE PASSWORD

The password in SPOOK can be **20-characters long** and is alphanumeric which means that it can accept alphabets, numbers and punctuations alike. In other words any of the 256 ASCII characters excepting the 32 control characters are accepted. Since each of the 20 positions can be filled by any of the 224 (256-32) characters, we get **224^{20} combinations** for the whole password. The user has these many passwords to choose from. If expanded the figure would look something like this 567,000,000,000,000,000,000,000,000,000,000 … i.e. a number with numerous zeros.

EXPLORING FURTHER…

The Password is not stored - The password specified by the user is not stored in any form. Remember, if a program stores the password, irrespective of the form it is converted to, the work of the hacker becomes very easy. He just has to get the password out of the data after figuring out the method by which it is being converted and stored. This defeats the strength of the algorithm or other security features of the software. SPOOK is totally safe in this regard. We welcome any tests to verify our claim. **How do we recover the data when the password is not stored?** Before we explain the answer to this question lets explore the key points of the algorithm a little more.

Lets say your data is "**ABCDEFG**" and we use the password "**x123**". Your encoded data may look something like this "**L?1,;43**". Now, what happens if we change the password? say from "**x123**" to "**x122**". The output changes completely to say "**3=-|acE**". This is **password-dependency**. Each password will produce a totally different output. The total number of possible passwords have been shown above and therefore the encoded form of your data has also these many possible outputs.

This becomes possible because the **password itself is the key** and is used in full as a parameter in the algorithm. The algorithm can be represented as $O = f(I,k,x)$ where O is the output data, I is input data, k is the algorithm constant and x is the password or key. To state precisely the password **x** is user specified parameter in the function and causes a variation in it. In simpler terms it can be stated that each password has its own algorithm.

What is **data-dependency**? This is users pride and hackers envy. We have stated above that the algorithm can be expressed as $O = f(I,k,x)$. Each data block, say a character, which is passed through this algorithm, is used to change the k parameter before it is used to encrypt the next block of data. For example, if "**A**" is encrypted to say "**T**" then before the next data block/character say "**B**" is encrypted the algorithm constant k is changed with respect to "**A**", the previous data block. The algorithm can then be stated as $O = f(I,f(k,d),x)$, where **d** is the last data block that was encrypted. This implies that the algorithm changes as the encryption proceeds and is therefore **data-dependent**.

HOW SECURE IS SPOOK??? (Continued…)

Another point is **pattern-traps**. Try encrypting a fixed pattern say "**AAAAABBBBB"**. Perhaps you would expect a pattern in the resultant encrypted data also. May be something like "**:::::ppppp" or "-----&&&&&",** but the data dependency of the algorithm defeats this. The algorithm varies after encrypting each block (data-dependent), after encrypting the first block say "**A"** to "**:"** the algorithm would be changed so that the next similar data block say "**A"** does not generates "**:"** again. The encrypted data may actually look something like this "**:i9;=-\ac2"** instead of the expected pattern previously shown.

You must have by now understood the role of password in the algorithm quite well so we come back to our question of restoration of data without the password being stored. The password specified is used to generate an algorithm. This algorithm is then used to start decryption. The data decrypted may be correct or incorrect depending upon the correctness of the password but **something would be decrypted**. After decrypting a small chunk of sample data the integrity of the decrypted data is checked through a checksum. If the checksum fails the password is deemed to be incorrect and the decryption stops with an appropriate message else its proceeds normally. So it is for you to remember the password, or else, go in for trying all the possible combinations and get back your data.

We now come to another important term - **Impossible**. Some understanding would be required on the part of the user in this regard. Impossible is more relative than absolute when it comes to encryption. We use the password as the key and it is possible to try out all the combinations and decrypt the data but perhaps it would take around 250 years if you punch each password yourself. This is impossible OR Lets say you program 5 machines and put them to this job so the time taken would be much less say 5 years or even less. But is this also practically possible? Would you or somebody else put five dedicated machines for 5 years to decrypt your data? This will also be termed as impossible…perhaps. **So it is just the amount of resources required and the feasibility of employing them, which determines the degree of impossibility. Your data is secured by this impossibility when you use SPOOK**. Remember, the password is the key and longer the password the stronger the encryption. The encryption goes up to 160-bit if you use a 20-character password. We recommend a **minimum length of 8 characters for the password**.

CONCEALMENT

This is a special feature, which sets aside SPOOK from other similar products. The encrypted data can be stored either to an existing non-overlaid DOS executable (.exe file) or to floppy diskettes. In either of the cases the concealment provides an extra layer of protection to the user. How?

The data is encrypted and stored in an invisible manner on the floppy diskette. The diskettes would appear and behave like empty diskettes. Any software can be used to check this fact. In fact if we copy something to such a floppy, the encrypted data inside it is destroyed as if it was not there at all. Any kind of warning or abnormal behavior by the floppy would provide the hacker with the first clue and place to start with.

Similarly, if a stand-alone file is used as a target, its content will appear as garbage to the outsiders. Without using Spook with valid password, no body will be able to make out anything out of it. It is practically impossible to even determine that the data is encrypted by Spook. So if a proper name of the target is chosen with a proper folder for ex: c:\windows\tmp00xxyy.001, it will be impossible for the hacker to even locate the data.

Data is therefore stored in a concealed manner and is **Invisible to Outsiders**. Remember, to decode your data, a hacker or outsider has to locate it first. This is where concealment comes in his way.