About

NOYBcryption™ is a system for encrypting and decrypting single files or whole hierarchies (folders) of files. And the encryption is done in such a way as to make it easy for you, the end user.

More importantly, **NOYBcryption** files are **plausibly unbreakable℠**. Unless an attacker has extreme computing power available, has infinite time or knows the pass phrase to decrypt with, your secrets are extremely safe. Visit our website to see a more in depth explanation of our system and why it is **plausibly unbreakable℠**.

NOYB - stands for "**N**one **O**f **Y**our **B**usiness" - which is what a secret tends to be! And **cryption** stands for "(En)/(De)**cryption.**

The name for this application is derived from its' ability to both encrypt and decrypt files which you want to keep secret. The files created can then be securely moved to where they are needed using standard email attachments, flash drives, etc. without regard to possible loss or interception. **NOYBcryption** requires NO public/private keys NOR any registration of such keys. **NOYBcryption** employs a "pass phrase", known only to the parties communicating.

So what is so special about NOYBcryption?

Good question. After all, there is all kinds of software out there that does encryption.

Well, **NOYBcryption** is "more better" than any of the others out there because it employs the encryption scheme known as OTP (One Time Pad) adapted by using 21st century software and algorithm (AKA stream encryption). Software algorithms and techniques are used to create pseudo random key material which will repeat no more often than once every 2^{19937}-1 ($4.315425 \times 10^{6001}$) iterations (that's a lot of iterations folks). To make the random key material cryptographically secure, an 8:1 hash reduction is performed on the random bit stream to create random numbers that pass the NIST "Die Hard Randomness Test Suite" with flying colors.

The NIST "Die Hard Randomness Test Suite" is a very famous and rigorous set of mathematical tests for randomness. The pseudo random generator algorithm employed by **NOYBcryption** has been rigorously tested with the NIST "Die Hard Randomness Test Suite" to be sure of this randomness. See our website for more information on the "Die Hard Randomness Test Suite".

The starting seed for the random number generator is composed of a part (½) derived from the pass phrase (private) and part (the other ½) which is very high quality random data (public). Seeded in this manner, the probability of a generator on the same machine producing the same sequence more than once is beyond *extremely* remote.

Another algorithm optimization is to compress the original data prior to actual encryption. Because of this reduction to a binary state of the original data, a further obfuscation of the original data occurs. When this zipped data is combined with the random key, the probability of ferreting out the original data is increased further toward impossibility.

The bottom line of this encryption scheme is that ANYONE intercepting NOYB encrypted files will be unable to break into them without knowledge of the “pass phrase” used to encrypt the files originally (OR extreme computing power beyond imagination).

The Software

There are two applications which comprise the original manifestation of the **NOYBcryption** system. They are:

NOYBcryption is capable of **both encrypting** and **decrypting** either an individual file or an entire directory of files and/or folders. This is the main application, licensed on a per computer basis, at a fixed price. Features include:

1. Pass phrase protected encryption & decryption using a pass phrase that should be changed on a schedule suitable to parties communicating
2. Upon encryption, creates a single, monolithic, encrypted file for easy handling and transmittal
3. Upon successful decryption, will re-create original file structure on recipient platform
4. (Will be) cross platform between Apple Macintosh & Microsoft PC (**Windows PC version not available yet**)
5. Authentication results automatically through use of pass phrase
6. No identifying information used in public portions of files created
7. No public key registration is required as this is NOT public/private key encryption, allowing for completely anonymous file ownership
8. Cryptographically Secure Pseudo Random Number Generator (CSPRNG) utilized with a repeat cycle of 4×10^{6001}, meaning any possibility of temporally or spatially meaningful brute force decryption borders on absurdity
9. On demand, will spawn a zipped **NOYBreader** (PC or Macintosh) to your desktop for subsequent transmittal to a recipient who does not have the **NOYBcryption** software
10. Always shreds ANY temporary files created on permanent storage, by the application, using random data

NOYBreader™ is capable of **decrypting** encrypted NOYB directories (*.noybd) and individual files (*.noybf). This is a reader application which may be distributed freely, under license, to anyone having a need to decrypt a NOYB file (with the correct pass phrase available of course).

1. Requests a pass phrase to use to decrypt a message and then saves the decrypted file(s) in selected location
2. A copy of this application is always instantly available to anyone with a licensed copy of the **NOYBcryption** application for either a Macintosh or a PC (Windows PC version not available yet) because **NOYBcryption** can spawn a copy of **NOYBreader** on demand.
3. Can also decrypt **NOYBmailer™** encrypted file attachments resulting in an RTFD file that may be opened by any RTFD capable program such as Apples TextEdit program.

These applications are pretty self explanatory in use. For more details and help, follow this link http://www.chssystems.com.

Why use NOYBcryption ?

It is based partly on OTP. And OTP is the ONLY provably secure encryption algorithm known.

Because a random key is simply combined with the “secret”, the secret also looks like it is random data. As a result, there is NO way to recover the secret once encrypted with random data. Since OTP is the only provably secure encryption know to exist, it is possible to encrypt a message using OTP and - unless the key can be recovered/ reproduced - the message will always be secure.

The same cannot be said about any other encryption method, including this one. They are ALL subject to being broken with back doors and/or brute force attack. Public/ Private key algorithms are “blessed” by the government and there is NO way to prove that they do not have some kind of back door and/or brute force attack vector that can be easily applied to any data traversing the internet.

OTP, combined with random keys that are never re-used in the same temporal space by the same parties, is unbreakable in the absence of the means to recreate the random key.

It is important to recognize that this software system **IS NOT an OTP encryption system**. Strictly speaking it is a stream encryption. It is **plausibly unbreakable℠** because one either needs the pass phrase OR one needs gargantuan amounts of computing power and/or infinite time in order to try sufficient numbers of combinations to actually have a brute force attack work. This is true because with our system, every attempt to break an encryption has a 1 in 20×10^{3000} chance of being successful.

Using the software techniques described above, and adapted into the OTP algorithm/ technique, the main deterrents to using an OTP are eliminated, as follows:

to use classic OTP	NOYB solution	NOYB method used
an encryption key must NEVER be used more than once	random number key sequence, for all practical purposes, is never re-used + key material is memory transient only, so it is NOT on ANY permanent media for accidental re-use	by employing 1) a user determined pass phrase, and 2) a very high quality random number generator, a seed for the OTP random number generator is set and the sequence is generated into memory where it is applied to data being encrypted
key material must be same size as material being encrypted	generate as much key material as required on the fly, anywhere an authorized pass phrase is available	keys will be generated from the same seed wherever required
key must be transported to both ends of a conversation	key transport is not necessary because the “secret” lies in knowing ONLY the pass phrase - the key will be generated where it is needed	because there is NO key transport required, the single worst problem of OTP (that is the compromise of key material), is totally eliminated

Your Pass Phrase

The pass phrase chosen to encrypt NOYB files lies at the root of the security afforded by the **NOYBcryption** system.

- **IF AN ATTACKER EVER FINDS OUT WHAT YOUR PASS PHRASE FOR A MESSAGE IS, ALL SECURITY IS LOST - YOUR MESSAGE/DATA WILL BE COMPROMISED**

- **EQUALLY AS IMPORTANT IS THAT YOU MUST NOT LOSE KNOWLEDGE OF WHAT YOUR PASS PHRASE IS. YOU WILL NEVER, EVER, EVER RECOVER YOUR ENCRYPTED DATA IF YOU LOSE YOUR PASS PHRASE**

There is no clearer way to state the above. It is key to the safety of your data. It is solely up to parties communicating to maintain a procedure for exchanging the pass phrase, changing the pass phrase, pass phrase change schedule, etc. Some absolutes about this are:

- Use a pass phrase larger than the minimum enforced 8 characters

- Use a mix of upper & lower case, numerals and other standard ascii keyboard characters
- Don’t just use the same pass phrase forever
- Perhaps have an agreed to, ahead of time, change schedule built into your pass phrase determination. Maybe it is the third word of the banner headline of your city or national newspaper, perhaps followed by some sequence number like hour sent. It is up to you - but it is EASY to have a pass phrase which changes regularly which is known ONLY to you and authorized parties
- Don’t use the obvious passwords related to either you or parties authorized to know the pass phrase, like birthdays, names, ages of kids, birth year, etc. - an attacker can guess these things
- Don’t ever write down your pass phrase OR any scheme you may have for creating your pass phrase updates
- Treat the knowledge of your pass phrase in a manner proportional to the secrets you are keeping. If your secrets are VERY secret, act accordingly with your pass phrase change protocols.
- Different pass phrase scenarios are probably a good idea for different uses of the **NOYBcryption** system. If the intent is to encrypt files for a period only long enough to effect transmittal, that requires one kind of scenario. If a file is meant to be kept encrypted for a long period of time prior to decryption, another kind of a pass phrase scenario might be in order. This aspect of pass phrase “generation” and “recovery” is very important to pay attention to.
- Whatever you do with pass phrases, don’t lose them ... at least until you have decrypted the message.
- Don’t compromise an old or retired pass phrase as the encrypted file may still be in the wild someplace (an internet email server for instance) with an attacker just waiting for you to accidently reveal that old pass phrase.
- All the above is why pass phrases belong only in your mind and in the minds of those you communicate with using **NOYBcryption**.

Limitations on the software

Our system is NOT meant to encrypt “large” volumes of information OR entire disk volumes. Other products in the marketplace might address this need. Also, alias files will **NOT** be followed, but they will, upon decryption show up as empty alias references.

NOYBcryption has been tested successfully with small, individual files all the way up to 100MB folder/file trees. If you have larger entities than say 100MB, please consider breaking these structures up into smaller pieces or using a different product.

The 100MB suggested upper limit cited above was chosen not because of any artificial limitation in our system but simply based on memory utilization and time demands observed on test computers. Going much beyond 100MB is up to you - but remember that results will vary depending on your system and limits inherent in that system.

Release Information

Release Version	Date	Change Description
1.0.0	Feb 13, 2008	Original release to beta.
1.0.1	Feb 22, 2008	1. Corrected library file purge mechanism to correctly shred files prior to deletion. 2. Added time elements to public key for MT seed in case somehow the random seed turned up twice. While extremely unlikely that /dev/urandom would ever return same 1,248 bytes, decided to add time to the seed so it is even more unlikely.
1.0.2	Feb 24, 2008	1. Add license menu item. 2. Remove references to NOYBmailer software. 3. Cleanup.
1.0.3	Feb 28, 2008	1. Make license reading forced upon first use and adopt a "Fair EULA" from the internet. 2. Setup **NOYBcryption** so it can spawn copies of the **NOYBreader** application, for either PC or Macintosh. 3. Work on this documentation.
1.0.4	Mar 3, 2008	1. Cleanup code. 2. Change pricing table. 3. Make license printable.
1.0.5	Mar 4, 2008	1. Correct registration code. 2. Work on this documentation.
1.0.6	Mar 12, 2008	1. Randomize "in-the-clear" memory objects 2. Fix aesthetics 3. Licensing lib based 4. Correct documentation
1.0.7	Mar 19, 2008	1. Aesthetics (icon).

Legalese

Please see the software license agreement visible from within the Help menu of all of our applications.