

Technology Overview
Strong Encryption
BackupEDGE™

The importance of standards-based data encryption.

Safe and Secure encryption is as much a strategy as it is a technology

Introduction to **BackupEDGE** Data Encryption

A major feature of **BackupEDGE** is the ability to protect archives containing critical client data by providing highly secure, standards based data encryption while maintaining all of the functionality users have come to expect of our products.

Commitment to Design Excellence and Client Protection

Safe and secure encryption is as much a strategy as it is a technology. Mistakes can compromise both security and data integrity. Before beginning engineering on this product, we considered the following:

- What kind of encryption key creation and management strategy makes the most sense?
- How do we make sure that safe, reliable encrypted backups (with verification) can be performed, even if the user chooses not to place the decryption keys on the system itself?
- How can we be sure that access to one old stolen, lost or discarded archive doesn't provide an easy attack method on newer archives?
- How do we minimize the effect of encryption on devices which perform hardware compression?
- How do we minimize the potential performance hit that encryption places on a system?
- How do we make sure that *none* of our current features and benefits are compromised, reduced or disabled when encryption is active?
- Can we be sure that users can change their encryption keys at will without compromising the ability to restore archives created with older keys?
- Will the resulting product be based on reviewable, open standards or simply "security through obscurity", which is not security at all?
- Will the resulting product be both FIPS (Federal Information Processing Standard) compliant and adhere to government export regulations?

Encryption is fully integrated into **BackupEDGE**, not just bolted on as an afterthought. As a result:

- *No* features are disabled when encryption is used.
- Users have multiple choices of key strategies.
- Key backup capabilities are designed into the product.

Microlite Corporation
2315 Mill Street
Aliquippa PA 15001-2228 USA
Tel: 724-375-6711

email: sales@microlite.com
web: www.microlite.com

© 2003-2024 by Microlite Corporation
All Rights Reserved

Technology Overview

BackupEDGE uses a powerful combination of symmetric and asymmetric algorithms to encrypt and decrypt data. It is completely standards-based, and the methodology is published here to assure users that standards are being followed and that no “back doors” or other security holes are in place. Users of encryption should be aware of the potential consequences of lost or stolen keys or pass phrases before utilizing this technology.

BackupEDGE encryption is completely standards-based.

Encryption is fully integrated into the product and performed at the file level.

No features are compromised or disabled when using encryption.

Encryption is fully integrated into the product and performed at the file level, providing the following benefits:

- only data that needs to be protected is encrypted.
- overall performance stays high as only critical files are subject to CPU intensive encryption.
- full compatibility with our bit-level verify, file checksum verify, indexing, quick file access and disaster recovery features is maintained.
- each encrypted file is pre-compressed using the powerful *zlib* libraries to ensure that no space is lost due to the inability of tape hardware compression implementations to compress encrypted data.

Optionally, a user may choose to encrypt an entire archive (except the file headers), although this is not generally desirable.

Archive Protection Methodology

For maximum security, each archive is encrypted with its own private, **256 bit AES encryption key** using the well documented Rijndael (pronounced Rhine-doll) formula. Separate, randomly generated keys for each backup (called the “session keys”) assure that access to multiple archives does not provide a useful method for attacking the encryption.

Further, each file is compressed before encryption and a random byte is inserted into each 15 byte block of compressed data, further thwarting attempts to attack the encryption based on the attackers’ potential knowledge of the pre-encrypted contents of one or more files on the archive.

The 256 bit encryption key for each session is created using a cryptographically strong, non-deterministic random number generator.

The Secret of the Keys

With data so powerfully encrypted using large, randomly generated keys, the natural question is “*How do we decrypt the data when we need it?*”! The answer is, we store the session key, which is both the encryption and decryption key, right on the archive. Before you ponder how silly it sounds to put the decryption key on the archive, we should point out that the session key is itself encrypted using powerful **RSA 2048 bit public/private key encryption**.

During product setup, a public and private encryption key pair are generated using the same random number generator previously mentioned. The public key is used to encrypt the session keys, and may be made public knowledge without compromising security. In fact a single public key may be placed on more than one system, which is especially useful in replicated site environments.

Microlite Corporation
2315 Mill Street
Aliquippa PA 15001-2228 USA
Tel: 724-375-6711
Fax: 724-375-6908
email: sales@microlite.com
web: www.microlite.com

BackupEDGE encryption protects archives, not systems.

It is a given that any UNIX or Linux user with **root** access can compromise a system in a variety of ways that don't involve archives, so this product should be viewed as a supplement to good system security.

The private key is used to decrypt files on restore. More properly phrased, the private key is used to decrypt the randomly generated session key from the archive, which in turn is used to decrypt the actual archive files.

The RSA keys are never placed on a data archive unencrypted. They **must** be guarded and archived separately by creating a *Key Archive*. New keys may be generated at any time, and any number of decryption keys may exist at one time.

Guarding the Keys

BackupEDGE encryption protects archives, not systems. It is a given that any UNIX or Linux user with **root** access can compromise a system in a variety of ways that don't involve archives, so this product should be viewed as a supplement to good system security.

During RSA key generation, one public and two types of private keys are created. *Standard Private Keys* are protected by UNIX/Linux system privileges. *Protected Private Keys* are additionally encrypted with a passphrase. After creating and archiving the private keys, the administrator may choose to remove those keys from the system with the following effects during restore:

- If a *Standard Private Key* exists for the archive in question, files are decrypted and restored automatically and transparently.
- If only the *Protected Private Key* exists, the administrator will be prompted for the pass phrase before the files may be decrypted.
- If neither private key exists, the user will be prompted to insert the appropriate *Key Archive* before the files may be decrypted.

During disaster recovery, the *Key Archive* must always be inserted.

Things To Consider

Developing encryption products requires a deep understanding of how encryption technology works *and* the many ways in which encryption can fail to protect data when improperly used. Because of the latter, it is important to remember what to look for when considering encryption technologies. Here are a few common mistakes to watch out for:

- Using only symmetric encryption technology usually means that each backup uses the *same encryption key*, providing an easier method of attack.
 - **BackupEDGE** uses a combination of symmetric and asymmetric encryption. It generates a new (symmetric) session key for each backup randomly, and transparently to the user. The user needs to be concerned only with managing the asymmetric key pair.
- Using only symmetric encryption also means that the encryption and decryption keys are identical, so the decryption key *must* be on the system at all times to permit unattended encrypted backups.
 - **BackupEDGE** uses asymmetric encryption for key exchange, which means that it separates the encryption key from the decryption key. Only the encryption key must be stored on a system to perform an encrypted backup. Further, it is easy to store a copy of a single encryption key on many systems, *while keeping the decryption key only on some, or none, of them.*

Microlite Corporation
2315 Mill Street
Aliquippa PA 15001-2228 USA
Tel: 724-375-6711
Fax: 724-375-6908
email: sales@microlite.com
web: www.microlite.com

BackupEDGE generates a new (symmetric) session key for each backup randomly, and transparently to the user.

This greatly reduces the risk of a compromised decryption key. (Recall that the encryption key can be made public without any loss of security.)

- Secure key creation is important. Simple manipulation of passphrases to generate a key, such as computing its MD5 hash, provides very little actual security. Users will tend not to use a hard to guess passphrase, making cracking simple *regardless of the encryption algorithm used*.
 - **BackupEDGE** creates all keys with a **non-deterministic, cryptographically strong random number generator**. (Optionally, the decryption key may be further encrypted by a human-supplied passphrase to protect it from casual observation. However, even if an attacker guesses this passphrase, it isn't useful *unless they somehow also get access to the hidden private key itself*.)
- Leaving keys in human-readable form or in insecure locations compromises security.
 - **BackupEDGE** decryption keys don't have to be on a system at all to allow encrypted backups. If you do choose to store the decryption keys on a system, then they are always protected by UNIX permissions, optionally hidden by a passphrase. Encryption keys, in contrast, can always be made public.
- Not having the ability for the archive itself to understand the proper required decryption key, or having restored keys for old keys overwrite newer keys, compromises restore capabilities.
 - Each archive label includes information about the key needed to restore it, so that **BackupEDGE** can automatically select the right key. If it needs to prompt for a passphrase, then it can provide a short, user-supplied description of the key and when it was created.
- Products that simply encapsulate an entire archive with an encryption filter can be dangerous. Potential problems are...
 - 1 Read-error recovery. Single byte errors could render an entire archive unrecoverable. There would be no easy way to sync back up with the encryption stream.
 - 2 Lost Quick File Access / Instant File Access.
 - 3 Long restore times increase dramatically. The entire archive has to be decrypted just to restore a single file.
 - 4 Poor system performance. Backup and verification time windows are greatly expanded.
 - **BackupEDGE** integrated encryption encrypts only the data you specify, and avoids all of these problems!
 - 5 Standard, hardware compressing tape drives would suffer from greatly reduced performance and capacity.
 - 6 The archive would actually be less secure! Because much of the data in backup archives is repetitive, dictionary-based attacks are possible even with access only to a single archive. Access to two or more encrypted archives could further enhance the feasibility of this attack.
 - **BackupEDGE** first compresses data, then inserts random bytes, before encrypting it. Every backup has a new symmetric key that is created by a cryptographically strong random number generator. These features enhance archive

Microlite Corporation
2315 Mill Street
Aliquippa PA 15001-2228 USA
Tel: 724-375-6711
Fax: 724-375-6908
email: sales@microlite.com
web: www.microlite.com

security while shrinking the space needed to perform a backup.

Simply encapsulating an entire archive with an encryption filter can be disastrous. **BackupEDGE** encryption is fully integrated at the file level.

As you examine other technologies, remember that if any *one* of these bullet items has been ignored, then your data is potentially at risk. If *more than one* of the deficiencies outlined above exists, the entire encryption strategy should be considered ill-conceived and discarded.

Network Backups

With encryption and network backups FTP-based backup Resource, the user can choose to:

- encrypt entire archives or parts of archives.
- encrypt the transmission using FTPS.
- encrypt both the transmission and the archive contents.

FTPS transport encryption alone does not require the encryption license.

Cloud Backups

Internet backups to Cloud Servers using the S3CLOUD Resource use the HTTPS protocol. These backups are made to sits like **Amazon S3**, **Google Cloud Storage** and **Wasabi**. The transmission is always encrypted regardless of whether the archive data itself is encrypted. If archive encryption is used, the user can choose to encrypt archives or parts of archives.

Summary

BackupEDGE fully integrates a robust combination of multiple encryption methods, data compression, and improved verification to assure maximum archive *security* while preserving storage *space* and *performance*. Peer review of the encryption methods used ensures that they are robust, *complete* and secure.

Scheduled nightly backups may now have their critical data completely secured (encrypted, backed up, bit-level verified) with no operator intervention. The operator assigned the task of rotating media does not even need login access.

Encryption has been an available option in **BackupEDGE** since 2003. It is enabled during the 60 day evaluation period of all demo/evaluation copies of **BackupEDGE**.

There is no sense spending thousands to protect your systems and your networks from intrusion, only to have someone walk off with an easily concealed tape, optical or flash drive, etc. containing all of your confidential data. Secure them safely with confidence and no compromises using **BackupEDGE** with the *Encryption Supplement*.