

**Introduction**

GMS Data Encryption protects certain aspects of some forms from unauthorized user access. Unlike GMS Security, which can be turned on and off, once data that is encrypted can never be recovered without the appropriate keys.

This document describes three basic processes:

1. Setting up master encryption keys (this only needs to be done once for each encryption group), and
2. Creating user decryption keys
3. Providing and un-providing keys when needed within GMS

*Hint: Any text you see formatted like this in blue is very technical and can probably be ignored if you don't understand it.*

**GMS Encryption basics**

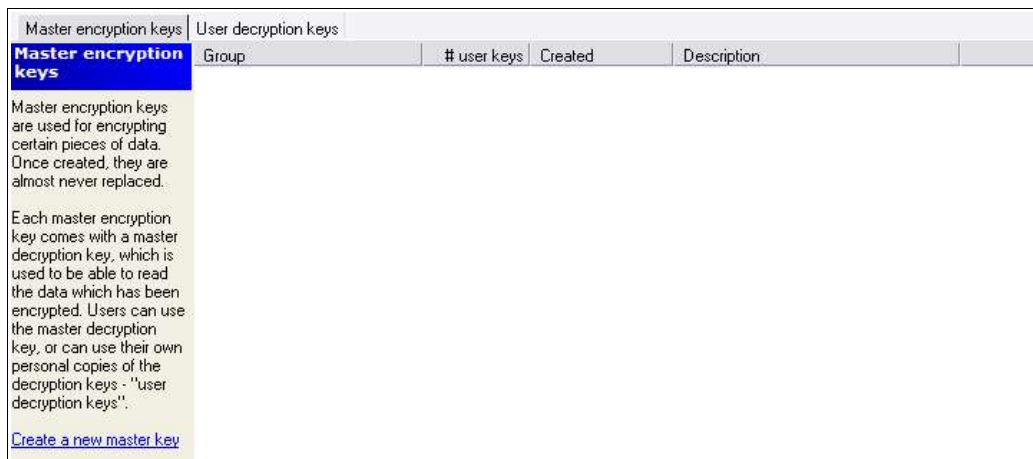
GMS uses a special form of encryption, called “Public key encryption”. It allows data to be entered and secured without the user being able to read that data later. Only those with the “private key” can read the data even though the “public key” is available to all users. The private key is then stored on a physical disk or USB thumb drive, not in the database itself.

For additional flexibility, “User decryption keys” can be created, specific to individual users. Each of these keys is protected by a password, and they can be revoked individually even if the user keeps the physical key.

It is important that the decryption keys – especially the master keys – not be stored on your computer.

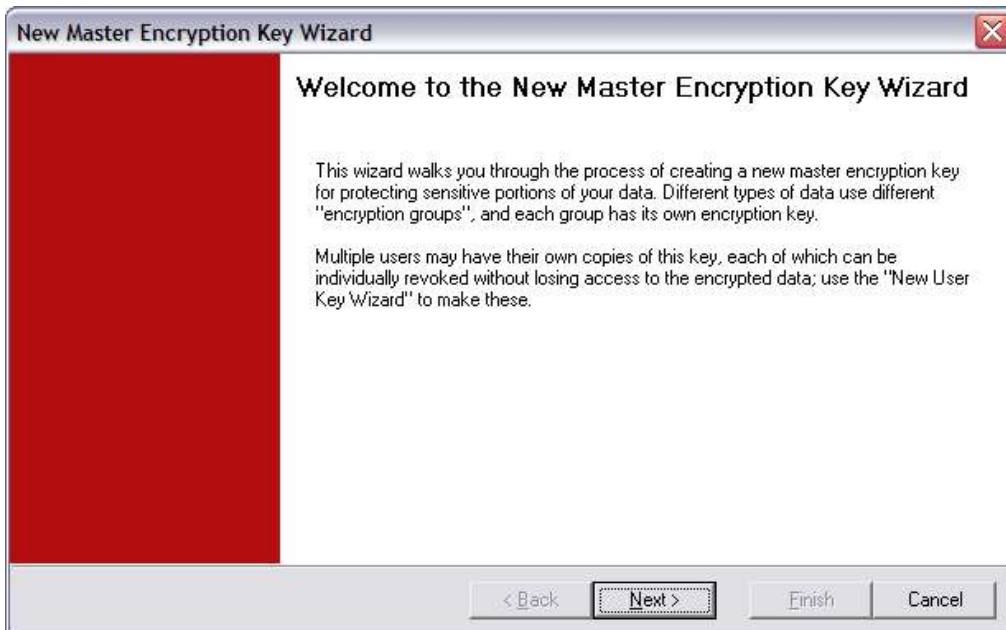
**Creating a master encryption key**

Open the Encryption Key Manager by clicking on the “Tools” menu, selecting “Tools”, then “Encryption key manager”. Each type of object being encrypted has its own master key; the most basic key is “BackgroundChecks”, which is used for storing information about volunteer and staff background check requests and results.



*Illustration 1, Encryption Key Manager*

Since the key that we need, “BackgroundChecks”. Since that key isn't here – and wouldn't be until you make it for the first time – we need to create that key. Click on “Create a new master key” to start the wizard.



Click on [Next] to specify the key to be created.



Illustration 2, Selecting the key to be made

Select the "BackgroundChecks" encryption group, and enter a key name if you want it called something other than the default. Click on [Next] to continue.



*Illustration 3, Saving the new master decryption key*

You should save your new key in two different places. The key that you're making here is a *master* key, which has no password associated with it and which cannot later be revoked. At least one copy of this key should go into a safe or a safe deposit box, since once data is encrypted with it, that data cannot be recovered without at least one copy of this key, or of a user key derived from it.

Click on [Next], and then on [Finish] to create and save the key (depending on the speed of your machine, it may take GMS several minutes to generate the key.)

Take your master encryption key and make a copy (or two) for a safe place. In the example, the file is simply `q:\BackgroundChecksMasterKey1.gpk`. You can burn this to a CD, or put it on a USB disk.

**User decryption keys**

User decryption keys work like the master decryption key – they are used to read stored, encrypted data – with two important exceptions:

1. They are protected by passwords
2. They can be individually revoked, even if you cannot get the key back from the user (for example, the medium that the key is stored on is stolen)

Note that the master decryption key can not be revoked. The best policy is for all users to have their own user decryption keys and for the master decryption key to be stored in a safe place, with at least one copy offsite.

Open the Encryption Key Manager by clicking on the “Tools” menu, selecting “Tools”, then “Encryption key manager”.

Master encryption keys	User decryption keys			
<p><b>User decryption keys</b></p> <p>User keys are copies of the master decryption keys, but associated with individual users. The master decryption keys cannot be revoked without loss of all of the data they are associated with.</p> <p>User decryption keys can be revoked, and each user can have his or her own copies. If a user leaves the organization, his/her keys can be revoked and made useless, even if they retain their copies of their keys, and other users' keys still remain valid and can be used for accessing encrypted data.</p> <p><a href="#">Create a new user key</a></p>	Group	User	Created	Status
	BackgroundChecks	BDOE	06/19/2005	Valid
	BackgroundChecks	JDOE	06/19/2005	Valid
	BackgroundChecks	TDOE	06/19/2005	Revoked
	BackgroundChecks	TDOUGH	06/19/2005	Valid

*Illustration 4, User decryption keys*

All of the defined user decryption keys are listed here. Each user ID may have only one user decryption key for each master key.

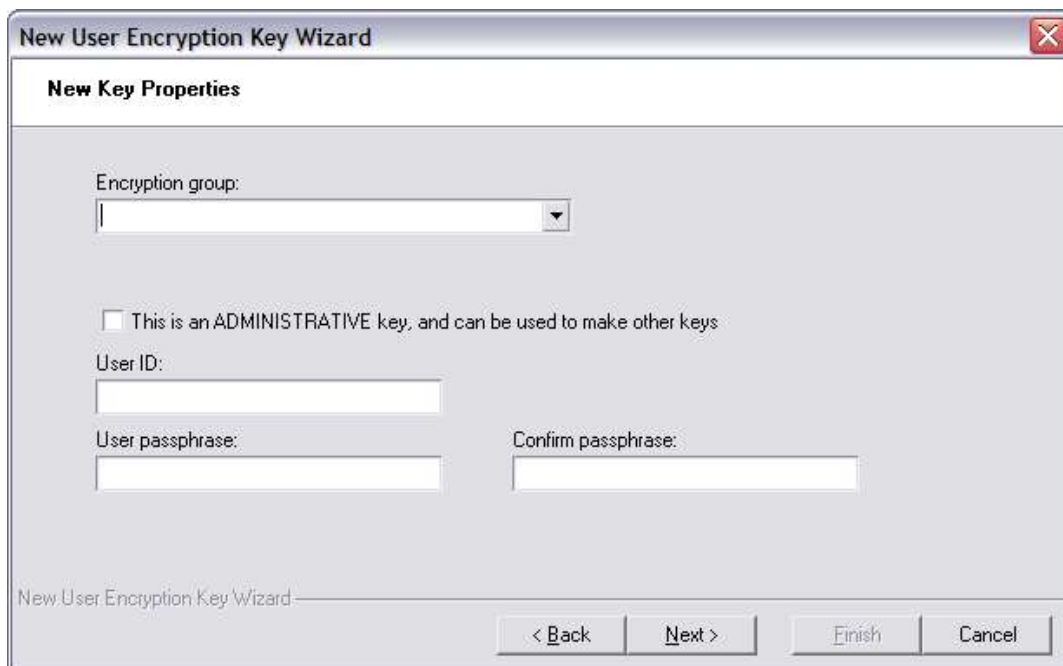
Note: User IDs in the key manager are not necessarily related to user IDs in GMS Security. GMS Security and GMS Data Encryption work well together, but neither requires the other and user IDs in one don't have to be the same in the other.

*Creating a new user decryption key*

Click on “Create a new user key” to bring up the wizard.

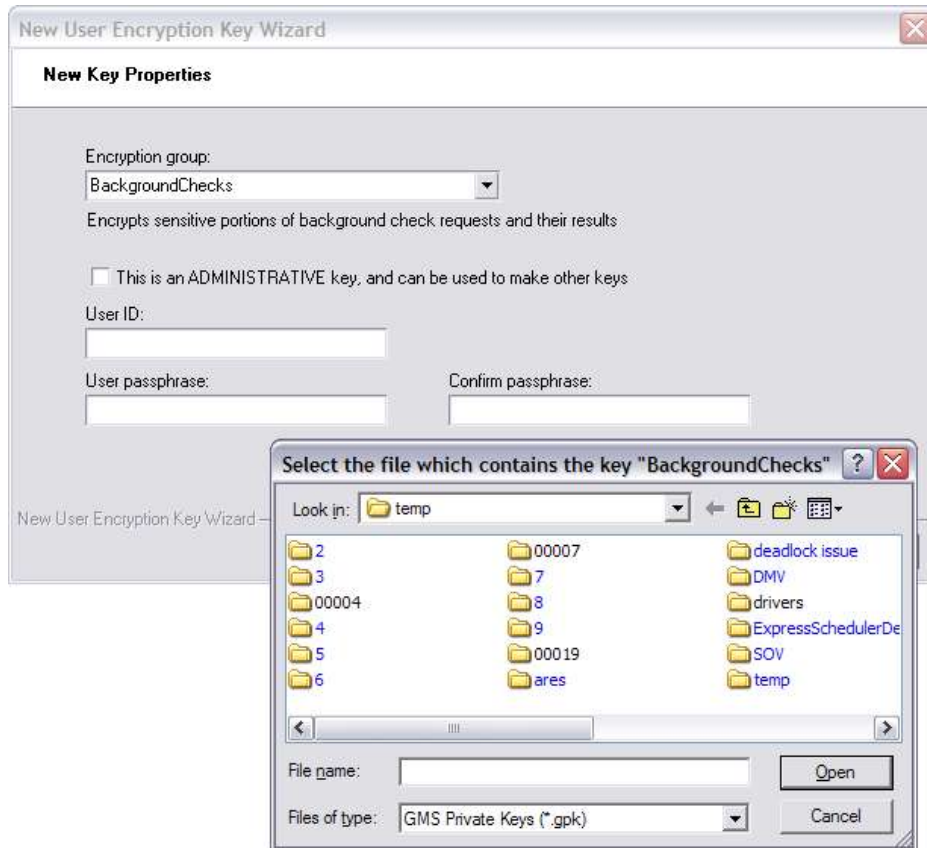


Click on [Next] to edit the key's properties.



Select the encryption group to make the key for, in this case, “BackgroundChecks”.

In order to make a new key, GMS requires that you provide either the master decryption key, or a user decryption key that has been marked as “Administrative”.



Select the file containing one of these keys, then enter the password if required.



Choose whether or not the new key can be used to make additional keys, enter a user ID and the user's passphrase. (The user can change his own passphrase later, if necessary.) Click [Next].

Choose where this user's key will be stored. Again, this should be an external medium like a USB key. Two copies can be saved in different places, and it can be easily copied later by copying the .gpk file it's stored in.



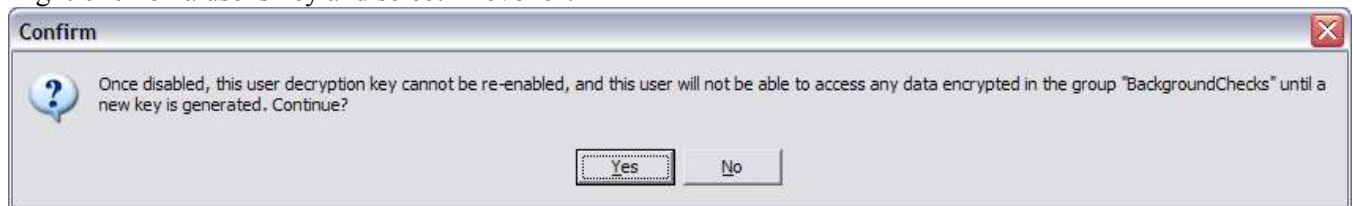
Click on [Next] and then on [Finish] to create and save the key.

*User decryption key notes*

User decryption keys should also be kept securely – never stored on the PC itself, for example, but are inherently more secure than the master decryption key since they are protected by passwords and can be individually revoked.

*Revoking a user's key*

Right-click on a user's key and select “Revoke”.



Clicking on “Yes” here will revoke the key and all copies of it, and a new key will need to be generated for the user if they are to access the data in the future.

*Changing a user decryption key passphrase*

Doing this requires the physical data file in which the key is stored, and the user's current passphrase. If both are not available, you'll need to create a new user decryption key from scratch.

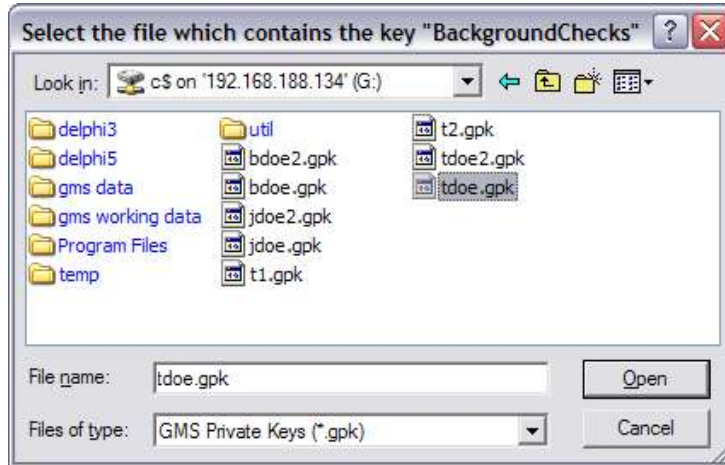
Right-click on a user decryption key and select "Change passphrase". Select the file with the user's key, and enter the current passphrase when prompted. Enter the new passphrase twice (for verification), click [OK], and then choose the file to save the modified key in.

Note that if the user has more than one key, either each one individually must be updated, or the file containing the key can be copied to the second location.

**Important note:** if a user's key or passphrase has been compromised, do not just change the user's passphrase. Since the key itself can be copied, and the passphrase is associated only with the copy of the key, the compromised key can be used to access data. Instead, revoke the user's key and create a new one. This will make the old key and all of its copies useless, regardless of how many times it's been copied.

**Providing keys within GMS**

When GMS tries to access an encrypted field, it will prompt you for a decryption key for that field. Once a key is requested, the “Key Manager” toolbar will appear on the bottom of the screen, showing the keys that have been provided, and those needed but not provided.



*Illustration 5, Prompting for the key "BackgroundChecks"*

If you give it a user decryption key, you'll also need to provide the password. If you get the error “Unable to load key: Invalid RSA Key”, you have either entered an invalid password or the key has been revoked.

After a key has been successfully provided, it shows in the Key Manager toolbar:

Keys provided: [BackgroundChecks](#)  
 Keys needed: (none)

**KEY MANAGER**

*Illustration 6, Key manager, with one key provided*

Clicking on a key name next to “Keys provided” will “un-provide” the key, meaning that GMS will forget that it has that key and will be unable to decrypt any data. Use this when you walk away from your machine to prevent anyone else from accessing what they should not.

If you are unable to provide the key, or click [Esc] when prompted for the file, the toolbar will look like:

Keys provided: (none)  
 Keys needed: [BackgroundChecks](#)

**KEY MANAGER**

*Illustration 7, Key manager, with one key needed*

Click on a key name next to “Keys needed” to have GMS prompt you for the key so that you can access encrypted data.

### How GMS encrypts data

This section is highly technical and is provided only as a reference for those trained in encryption, and for the curious.

GMS uses two-phase encryption on encrypted data, similar to PGP. For each record to be stored, a random 192-bit AES key is generated, and the data is encrypted with that key. The AES key is then encrypted using RSA, and prepended to the AES-encrypted data. This gives the performance of AES (very fast), with the public/private key abilities of RSA (very slow).

Every record uses its own AES key, so each record must be decrypted individually using the private key. Within the record, all elements which are to be encrypted in the same encryption group are bundled together and encrypted as a group, to minimize the number of RSA encryption/decryption cycles.

User keys are composed of three elements:

1. The user's passphrase, which is never stored
2. A random user code, generated for each user ID for each key, stored in the database
3. An encrypted version of the decryption key, which is stored on a USB key or other external medium

GMS encrypts the master decryption key with a combination of the user's passphrase and the user's database-stored code. When the user tries to use his key, GMS asks for the passphrase and looks up the user's code; if the two are provided correctly, GMS is able to decrypt the encryption key and use it.

When a user's key is revoked, that database-stored code is removed. Without it, GMS cannot decrypt the user's key, making the key useless.

Note: GMS' encryption is intended to hide the data itself, not the fact that data is present or missing. Empty fields are not stored, and the size of each encrypted bundle is directly proportional to the fact the size of the source data. It may be possible to determine that detailed information about an individual is stored, and therefore that history exists. From a large record size, someone may – correctly or otherwise – infer that negative data is present. The actual contents of the data, of course, remains secure.

### User passphrases

The user's passphrases are never stored in GMS or in the keys themselves. User key passphrases can be changed on individual keys, but only by the user or by someone who knows that user's passphrase. If the passphrase is lost, generate a new user key from scratch. No data is lost in this process, of course, since the user key is derived from the master key.