ESPC
Email Sender & Provider Coalition

MD5 Encryption of Suppression Lists

Frequently Asked Questions

(Companion to ESPC Document on Adoption of MD5 for Suppression List Encryption)

What does it mean to support MD5 encryption of suppression lists?

- You need to be able to provide MD5-encrypted suppression lists as output
- You need to be able to accept MD5-encrypted suppression lists as input (e.g. from an affiliate, or from a new customer bringing along their suppression list, etc.)

The implication of receiving MD5-encrypted input is that **all internal uses of suppression lists must be prepared to handle MD5-encrypted entries.** For those who continue to use clear text suppression list entries internally (and in general you will want to retain clear text entries for internal use), this means that **your system will need to be able to deal with entries of mixed type**.

How do I deal with entries of mixed type?

Since MD5 is a one-way encryption algorithm, the only way to deal with entries of mixed type is to encrypt any clear text entries prior to comparison. The type resolution could be done in real time, or more likely you would cache (temporarily or permanently) the MD5-encrypted values of your clear text entries to avoid real time encryption/decryption (i.e. you could do the MD5 encryption the first time you touch a particular entry, and then store the value so the encryption doesn't need to be repeated).

Do I need to use MD5 encryption for all suppression list transactions?

No, the requirement is to *support* MD5 encryption for suppression lists (as both input and output), not to use it for all suppression list transactions across the board. It won't be practical to require it for all transactions until there is broader industry adoption, because it could prevent compliance with CAN-SPAM: destinations that don't support MD5 would be unable to import data from anyone providing suppression list data solely in MD5 encrypted format.

Why do I have to normalize email addresses prior to encrypting?

The algorithm specified in the ESPC MD5 Suppression List Encryption document normalizes both the local part (the part before the '@') and the domain part of the email address to lower case. The local part is in fact case sensitive according to the specification (section 3.4.1 of RFC 2822 allows for lots of characters in the local-part of an email address, including upper-case characters), but some deployments nevertheless treat it as case insensitive. Thus the presence of a mixed case local part does not necessarily imply that the mixed case is significant: many deployments preserve the mixed case in email headers even when they normalize it internally because it is considered more aesthetic or user friendly. Because of the nature of suppression list checking, it is considered safer to always normalize and risk the occasional extra hit (when mixed case local parts are actually distinct but match after normalization) than to risk missing a match and violating CAN-SPAM. See http://tools.ietf.org/html/rfc2822#section-3.4.1 for all of the variations and parameters of the local-part of an email address.

Do I have to implement the MD5 encryption myself?

No, in general the practice when using encryption is to make use of 3rd party libraries. In general you will be able to find both open source and commercial libraries available.

How does this help? Can't someone malicious just decrypt the encrypted suppression list entries?

MD5 encryption, or any encryption algorithm suitable for use on suppression lists, is a one-way algorithm. That means that once something is encrypted with MD5, there is no direct way to extract the original text. (Even if the algorithm is cracked, the cracking won't guarantee that the regenerated text actually matches the original.)

If you need to retain internal access to clear text versions of your entries, you should make sure that you store MD5 encrypted versions alongside your existing clear text versions, and that you don't overwrite the clear text with the MD5 encrypted values.

Why MD5 rather than some other algorithm?

We are requiring MD5 encryption as a base in order to ensure interoperability both among ESPC member implementations and with existing implementations.

Although it has been demonstrated that MD5 can, in fact, be cracked or reversed, the cracking is more useful in an attack on password encryptions than on suppression list encryption. Cracking or reversing takes a moderate amount of computational power and the result is not guaranteed to match the original data, so it's unlikely to be particularly attractive to email list abusers.

Other algorithms may be supported in addition to MD5, if desired, and; in future the industry may change the default to a stronger algorithm. If you do choose to support additional algorithms, it's important to make sure your implementation allows for a tag to identify which algorithm is being used, since comparisons will only be valid if the same encryption algorithm is used for all items being compared.

When must my implementation be completed?

The ESPC encourages members to target September 1, 2009 for implementation and hopes full adoption will be completed by the end of 2009. Contact Jim Campbell if you have any questions.

How must I demonstrate my compliance when my implementation is complete?

Members will be required to make a public statement on their website that advertises your support of your support of MD5 encryption for suppression lists.

Because business models vary, the exact implementation of this website notice may differ from member to member. ESPC staff and the Tech Committee co-Chairs will work with members to address questions about individual implementations.

Regardless, you should contact Jim Campbell and provide him a pointer to the page on your website that advertises your support of this requirement.