ESPC
Email Sender & Provider Coalition

**BEST PRACTICE GUIDELINES FOR ESPs:**

**Adopt MD5 one-way encryption support for suppression list management**

**Issue**: Suppression list abuse is a major issue for ESPs, advertisers, and ultimately consumers. LashBack data indicates that at least 2.2 million sending IPs sent mail to suppression lists in the last 30 days. The impact on mailing and brand reputation can be significant. The impact on inboxes of consumers is significant.

**Proposal**: The ESPC recommends to members and to the industry at large that MD5 be supported as a best practice for sharing suppression lists:

1. ESPC members should allow their customers to download subscriber and suppression lists in MD5 one-way encrypted format.
2. ESPC members should support uploading of suppression lists in MD5 one-way encrypted format.
3. ESPC members should teach their customers about the benefits of using MD5 over plain-text distribution of data.
4. ESPC members should list their support of MD5 for distribution of data on their websites and any best practices documents they produce about data sharing.

**Reasoning**: Typical methods for sharing data "securely" include the use of public-key encryption or passwords on a ZIP file, but, those methods still provide a plain-text human readable version of the data after the password is applied. With MD5, there is no way for the recipient of the data to send email messages to the list – the data can only be used for compliance purposes and not for mailing.

Supporting the MD5 format as a suppression list distribution method will enable ESPC members to support clients that are concerned about suppression list abuse and will not share their suppression lists in plain-text.

Using MD5 eliminates the risk of accidentally sending email to a marketer's suppression files and greatly reduces the risk of them being stolen and abused by a person in the chain of custody of the suppression list.

**MD5 background:** MD5 (Message-Digest algorithm 5) encryption is an industry standard that has been used for years to protect passwords and verify that downloaded files have not been corrupted. An MD5 hash is typically expressed as a 32-character hexadecimal number that looks like this: `9e107d9d372bb6826bd81d3542a419d6`

MD5 is a one-way hash process - once an email address has been turned into an MD5 hash it cannot be turned back into the original source email address. But because the MD5 hash is consistent for each email address, two lists of MD5-hashes can still be compared with each other to determine if there are any matching records. This allows an advertiser to distribute a list of MD5 hashes that can be used by an affiliate or publisher to scrub their list – but without ever disclosing any real email addresses!

Obviously this is safer than distributing the email addresses in plain text because it prevents against human error, accidents, theft and fraud. By using MD5 instead of plain text, advertisers can be 100% confident that their unsubscribe list will never accidentally be sent an email message, will never be exposed to a third party, and will never be stolen or abused.

The technical details of MD5 can be found in RFC 1321:
http://people.csail.mit.edu/rivest/Rivest-MD5.txt

Many ESPs currently support MD5, including ESPC members. Additionally, more ESPs plan to build MD5 support into their solutions by the end of 2008.

Because MD5 is basically numeric in nature and fixed in length, it can actually be faster for your systems to clean mailing lists if they are stored in MD5, which could result in increased operational performance for your mailing campaigns.

**Why MD5 over other one-way encryption methods?**

Although other algorithms like SHA-1, SHA-2, or SHA-256 could also work, the ESPC is recommending one standard, currently MD5, for consistency/interoperability reasons and faster adoption across all member companies. However, since MD5 is now considered breakable (see section on Threats below), if ESPC members can support a stronger algorithm they are encouraged to investigate the needs of their customers and support those other methods in addition to MD5 when they can.

**Standardization of data for MD5 hashing**

Preparing email addresses to use MD5 hashing requires following a few conventions. Since MD5 will create a unique checksum for different variations of a valid email address we need everyone to make the proper transformations to their data before running it through an MD5 function, both on the distributing side and on the receiving side of this data transaction.

All whitespace characters should be removed from the message (including newline characters) All characters should be transformed to their lowercase equivalents[1].

**Examples**
**Incorrect:**
MD5('Test@yahoo.com') = **b1ed60d5f3d3b540ef48f0b276e68e55**
MD5('  test@yahoo.com') = **bd6f5c7193bcb1d5bc08a3801223a984**

**Correct**:
MD5('test@yahoo.com') = **88e478531ab3bc303f1b5da82c2e9bbb**

[1] **Note:** The above algorithm 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 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.

Once a list of addresses has been converted to MD5 format, the addresses can then be compared against other similarly transformed lists.

**Hashing Threats**

The most common form of hashing used today is MD5. While MD5 is better than plain text, MD5 is an outdated hashing algorithm and can be "cracked" using a typical personal computer. Newer hashing methods such as SHA-256 provide greater levels of security and require massive amounts of current computer resources and long amounts of time to crack.

Another threat against any form of hashing is a brute force attack where the attacker collects millions of email addresses, hashes them, and then compares the resulting hashes to the suppression file. The attacker would not gain any new email addresses but would be able to gain more information about the email addresses he already had, such as which are more likely to be active or purchase a particular product. There is no record of this attack being used in the wild.

Reasonable precautions should also be taken in the transfer of any data. Data should be transferred using security appropriate to your needs and to the expectations of your customers. Storing an unsubscribe file, even in hashed form, on an FTP site that allows anonymous login may for example be considered insufficient security for the data transfer.

The best security is always not to share your suppression list at all, or at least to restrict its distribution on a need-to-know basis.