M3AAWG
MESSAGING MALWARE MOBILE

Messaging, Malware and Mobile Anti-Abuse Working Group
M³AAWG DKIM Key Rotation Best Common Practices

December 2013

Executive Summary

DomainKeys Identified Mail (DKIM) is a standardized process that has proven to be a highly effective means by which a receiver can verify that the signed fields of an email have not been modified in transit. However, to minimize the risk of active DKIM keys being compromised, they should be changed frequently. This is a practice known as "key rotation." This document discusses why keys should be rotated, how frequently they should be rotated, and suggests the best common practices for doing so.

DKIM keys should be rotated at least every six months. Doing so reduces the risk of active keys being compromised, either by attackers cracking or stealing them. Frequent rotations standardize the rotation process. With a standardized process, the institutional knowledge will be available in case an emergency compromise requires an out-of-cycle key rotation.

Table of Contents

1. Problem Statement

The use of DomainKeys Identified Mail (DKIM) enables an email receiver to verify that the signed portion of a message has not been modified in transit. This is also useful in verifying that a message is associated with a specific sending domain and is accomplished by using a pair of public/private encryption keys. The private key is used to compute a signature based on the content of the email and/or some of the message headers (present or not). The public key, published in the DNS, allows receivers to verify that the associated private key was used to create this signature. (Primary reference RFCs: 6376, 5585, 5863)

Given, then, that DKIM relies on published public keys that are available for inspection, they are also a target of attack. As such, care must be taken to mitigate the risk that public keys can be compromised by minimizing the amount of time they are in active use. The frequent replacement of older keys with newer ones (a practice known as "key rotation") is an effective defense that should be included in any operational plan related to DKIM.

The length of a DKIM key is directly related to the amount of time taken to crack the key using current mathematical approaches (e.g., using matrix factorization). For example, keys that are 512-bits long take significantly less time to crack than keys that are 768-bit. Keys that are 1024-bits long (the current recommended length) are currently out of reach for general purpose computer systems, and 2048-bit keys are considered to be immune to cracking in today's computing environment.

While longer keys are harder to crack, the private (a.k.a. "signing") key is still vulnerable to being stolen if an attacker is able to compromise the system in which it is stored. Minimizing the amount of time a key pair is in use ensures that, even if the key is compromised, the damage can be contained to a reasonably short period of time. Further, with frequent rotations, an organization will have the ability to react more quickly and replace the active keys should they become compromised.

This document is written to address many of the common issues relating to key rotation that organizations face when deploying DKIM for signing. This guidance takes into account the complexities faced by many organizations, including the fact that many have multiple email streams, some with delegated subdomains, or streams sent on their behalf by third parties. Ensuring DKIM key rotation across all the streams can be a daunting task, although given our reliance on email as a trusted means of communication, it is one that should not be ignored.

2. Period of Rotation with Justification

According to the article "How a Google Headhunter's E-Mail Unraveled a Massive Net Security Hole" published by Wired magazine on October 24, 2012, mathematician Zachary Harris said that it only took him approximately 72 hours (and about $75 using Amazon Web Services) to crack a 512-bit DKIM key. While he was unable to crack larger keys due to a lack of access to sufficient computing resources, he points out that nation states do have the means to crack 768-bit keys. Further, it is worth noting that it was over a year ago that he cracked the keys and, given Moore's Law, it would take less time and resources to do today.

While it is incredibly important to use strong keys, it is also important to ensure they are rotated frequently. The longer a key is available, the more opportunity there is for an attacker to crack and abuse it. The primary goal of frequent key rotation, then, is to minimize the time during which a key can be cracked and during which compromised keys can be exploited.

A related goal is to ensure institutional knowledge related to key management is maintained. As was pointed out in the Wired article, Google was able to upgrade its keys within a couple of days while it took months for other companies to follow suit. The takeaway is that companies with an efficient process for managing keys are able to react much more quickly when necessary.

In this light, frequent key rotation can be viewed as a useful drill to ensure smooth performance when needed. Each time keys are rotated, institutional knowledge is shared in a way that supports:

- **Cross-Department Effectiveness** – This is especially important when, for example, the department responsible for managing DNS is not the same as the group responsible for email operations.
- **Cross-Vendor Effectiveness** – Given that many companies rely on third party vendors to send email on their behalf, these vendors also need to be incorporated into the process. Long-time vendors are reminded of the process with each rotation, while new vendors are brought up to speed.
- **Conformance** – Advance planning to rotate keys supports the ITIL, ISO/IEC 20000, and similar best practices. For example, frequent, planned key rotations could be classified as a "standard change" vs. an "emergency change" that incurs additional operational overhead.
- **Automation** – While rotating keys can be managed manually, frequent rotation will lend itself to the development of tools to help automate aspects of the process. Even simple automation, such as scripts created to handle the generation of new keys, helps to reduce mistakes and save time.
- **Redundancy and Bench Depth** – Ensuring that multiple people and departments are involved in the process minimizes the reliance on specific individuals. When only a few people understand how to rotate keys, and they do so infrequently, the risk of losing necessary knowledge is increased.

Given the realities that most operational teams do not have infinite time and resources, how frequently should DKIM keys be rotated? The answer to that question will largely depend on the complexity of any specific organization, the time it takes to rotate their keys, the perceived value of the protected email stream(s) and the company's view of acceptable risks.

Members of M^{3}AAWG explored various rotational frequencies and determined that DKIM keys realistically should be rotated twice a year to balance the risk of compromised keys and operational effort. This takes into account the potential of keys being cracked or otherwise compromised, plus the need to institutionalize knowledge about the key rotation process. This recommended frequency addresses companies with complex email flows and a number of third-party dependencies, allowing them the operational flexibility to maximize effectiveness. Other, less complex mail flows can rotate keys more frequently. Each company should decide the specific dates that work for them. For example, rotating keys each April and October is a reasonable spread for many online ecommerce companies and avoids making changes during heavy buying periods.

It is important to understand the various factors involved when deciding on the frequency that works for a specific organization. While a more frequent rotation (e.g., quarterly) would lower the risk of a key being cracked, operational realities may prevent such an aggressive schedule. Similarly, a longer frequency (e.g., yearly) not only increases the risk of a compromised key, but also diminishes the value of the institutional knowledge gained.

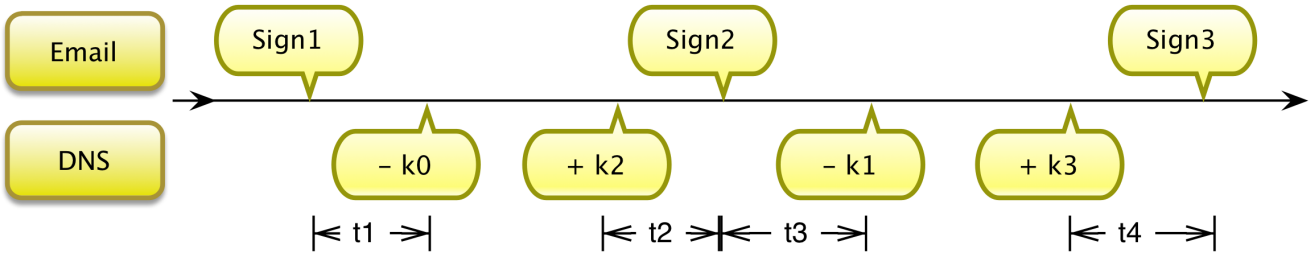
3. Operational Notes

In many organizations the mail administrators may be different from those who manage the DNS. Similarly, sub-domains may have been created that third parties can use to send mail on behalf of the organization. The sub-domain can either be controlled by the organization or by the third party. This requires appropriate coordination with all parties, especially considering the generation and use of DKIM keys. Since it is not recommended to share the same DKIM keys across several entities, care should be taken when considering third-party integration.

NOTE: *See Section 3.1 for specific considerations for third-party sending.*

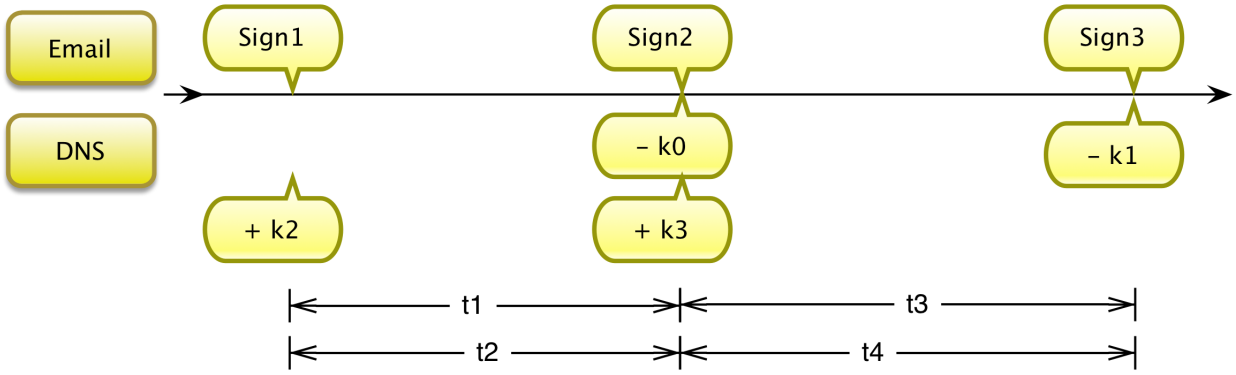
In order to ensure the DNS configuration is prepared to support large DKIM keys (e.g., > 1024-bit), it is necessary to verify that the DNS servers can support long text records. This means that the DNS servers will have to support EDNS0 or TCP-based record fetches.

Identifying a timeline and processes for each entity will allow smooth transitions at each rotation period. It is recommended to consider how to map the key activities (e.g., key generation and deprecation) and schedule recurring reminders for each operation in your operational calendar.



In the above timeline, the "- k" entries indicate a DKIM key being deprecated, while the "+ k" indicates a key being added for signing. Note that you need to allow enough time for DNS propagation of the new keys (t2 and t4) before using them, and you need to leave enough time for email to be received at MTAs before retiring keys (t1 and t3).

To simplify the process, you can choose to align some of the key activities. In this way, you do not have a key rotation that requires steps on several days but all operations happen on the same day, as shown below:



DKIM signatures support the specification of an "expiration time" (i.e. the "t=" value in the signature). This can be useful as an added security to ensure the signatures in email do not outlive the key rotation period. However, you should take care to ensure that rotation is guaranteed to happen before the expiration of the signature.

NOTE: *It is worth reiterating the recommendation that scripts be developed to simplify the automation of this process.*

3.1 DKIM Key Rotation in Third-Party Sourced and Delegated Domains

NOTE: *This document does not recommend that third parties initiate key rotation for their customers, but that they educate their customers to do key rotation in accord with this document.*

The key rotation request must come from the organizational domain owner; in many cases this is the customer. Once the request is received, the third party and the customer can use these guidelines to perform a smooth key rotation.

3.1.1 Key Delegation via Domain/Subdomain

The organization holding the domain creates the key pairs and publishes the public key in their main domain or in a subdomain, as appropriate. The private key is securely transmitted to the third party (e.g., via encrypted email using GPG or using a secure FTP server such as FTPS). This method has the inconvenience of requiring recurring coordination between your organization and the third party.

NOTE: *Never send a private key in email without it being encrypted or via other means that could introduce the risk of the key being observed in transit or otherwise stolen. Private keys should never be saved locally.*

An alternate approach is for the third party to generate the public/private key pair and provide the public key to the customer for publication in the organization's DNS. This approach avoids the need to transmit the private key between organizations. In this case, however, the customer should confirm that the generated key conforms to the customer's selected key generation parameters (e.g., key size), and there is mutual understanding as to when this key will be rotated out of production.

3.1.2 Key Delegation via Delegated Subdomain

With this process, the customer delegates a sub-domain to the third party which creates the necessary entries. This method is flexible but the delegation of a sub-domain to the third party may add risk that is unrelated to key rotation. For example, the third party could add a DMARC entry to the sub-domain that overrides the policy set by the main domain. It is therefore important to audit all delegated domains.

3.1.3 Key delegation via CNAME

Another method is to delegate keys to the third party by using CNAMEs. To do this, the domain holder creates three entries (key_1, key_2 and key_3) in the main domain that point to three entries at the third party. These three entries allow the third party to use one valid key while rotating the others. When "key_1" is expired, "key_2" is used to sign domains and "key_3" is the next in line to be used.

Example:

Consider the case where "example.com" is the domain to sign, and "acme.com" is the third party.

key1._domainkey.example.com CNAME key1.example.com.acme.com
key2._domainkey.example.com CNAME key2.example.com.acme.com
key3._domainkey.example.com CNAME key3.example.com.acme.com

key1.example.com.acme.com TXT "v=DKIM1; p="
key2.example.com.acme.com TXT "v=DKIM1; p=ADfe34556...."
key3.example.com.acme.com TXT "v=DKIM1; p=A783Fg4556...."

Note: *See Section 5 to understand why you can limit the number of delegated keys to three while still allowing normal rotation.*

3.2 DKIM Selectors Are Designed for Rotation

The intent of DKIM selectors is to enable key rotation. In order to ensure they can be used for this purpose, they must not be used for any other purpose.

For example, DKIM selector names should not be used to identify separate mail streams as they will change when the keys are rotated. Any internal business unit, email service provider or other third party relying on selector names beyond key identification must modify their methodology to avoid complications and unforeseen consequences.

NOTE: *If you are asked to enter a selector name into a form for some reason (e.g., feedback processing), common practice is to enter an asterisk (i.e. *). This will indicate that your DKIM selector names are not static and should be expected to change.*

4. Preparation and Readiness

4.1 Inventory

Identify all the mail streams being sent from your organization. This may include transactional messages automatically generated by a product sent to a customer, periodic marketing campaigns, or conversational messages sent from employees or other supported users.

To address the various mail streams, you will need to identify who manages each domain and who is in charge of the sending of mail via each stream. Each of these is a key stakeholder who needs to cooperate in generating, publishing and using DKIM within the mail system. While it may be easy at a small or medium-sized enterprise to identify these stakeholders, large organizations may have multiple business units operating nearly independently, each with their own mail streams that need to be identified.

Mergers and acquisitions should be audited to understand their mail streams and plan how to manage the key changes accordingly.

A useful tool in helping to identify email streams is the feedback reporting provided by the use of Domain-based Message Authentication, Reporting & Conformance (DMARC). Specifically, by publishing a DMARC "p=none" policy and associated Aggregate Reporting (RUA) address in their DNS, an email sender will receive daily reports about what mail is being sent (and purported to be sent) by the organization, including which messages have valid DKIM signatures and which do not.

4.2 Discuss Objectives and Policy with All Parties

You can use this document as a basis to discuss the rationale of DKIM key rotation within your organization. Share the plan, process, and timeline. It is important that everyone involved understands the value of protecting the email channel by regular DKIM key rotation. No one, including third party vendors and those responsible for the relationships with them, should be surprised when it is time to rotate keys.

4.3 Key Selector Naming Scheme

Define a naming scheme for the DKIM key selectors that is both meaningful for forensic analysis and is sufficiently random so the keys cannot be easily guessed by browsing the DNS.

NOTE: *The selector naming scheme should also be designed to mitigate the risk that attackers can easily predict the names of future selectors and retrieve the associated keys. See Section 5 for a description of the process for publishing keys for future use.*

Organizations with many different email streams may find it useful to prepend the name with the department responsible for the selector used by their stream. Including the length of the associated keys can also help with troubleshooting. A naming convention using rotation dates can help keep selectors ordered.

A suitably unguessable suffix added to the naming convention would be a series of random characters. Assuming an attacker guesses that the suffix is a random sequence, repeated probing by the attacker of the DNS looking for the right sequence could be detected and appropriate action taken (e.g., changing the naming scheme and rotating keys early).

An example using a random suffix may be: "sales-201309-1024-da7emuf9." This example indicates that it belongs to the "sales" email stream, is intended to be rotated into active duty in September 2013, and references a 1024-bit key. The random suffix is intended to mitigate the risk that the key could be guessed in advance.

Another option for a suffix that is hard to predict, yet also contains useful information, may be to use an epoch timestamp when the selector was created. Although trivial inspection will reveal to an attacker that an epoch timestamp is being used, repeated probing of the DNS by the attacker looking for the right timestamp could be detected.

An example using an epoch timestamp may look like: "sales-201309-1024-1377880393." This example indicates that it belongs to the "sales" email stream, is intended to be rotated into active duty in September 2013, references a 1024-bit key, and was created Fri, 30 Aug 2013 16:33:13 GMT.

Each organization needs to evaluate the tradeoffs between the anticipated risk of selector guessability and encoding useful information in the name.

NOTE: *The reuse of selector names is strongly discouraged.*

4.4 Stakeholders

Identify the internal stakeholders, email administrators, DNS administrators or the internal customer/technical support personnel as well as the customer and technical support staff at any third party vendors responsible for email streams. Identify the relevant processes and how to initiate modifications to the email stream; e.g., do you submit a support ticket, make an official request, send an email, define a contractual agreement, etc.?

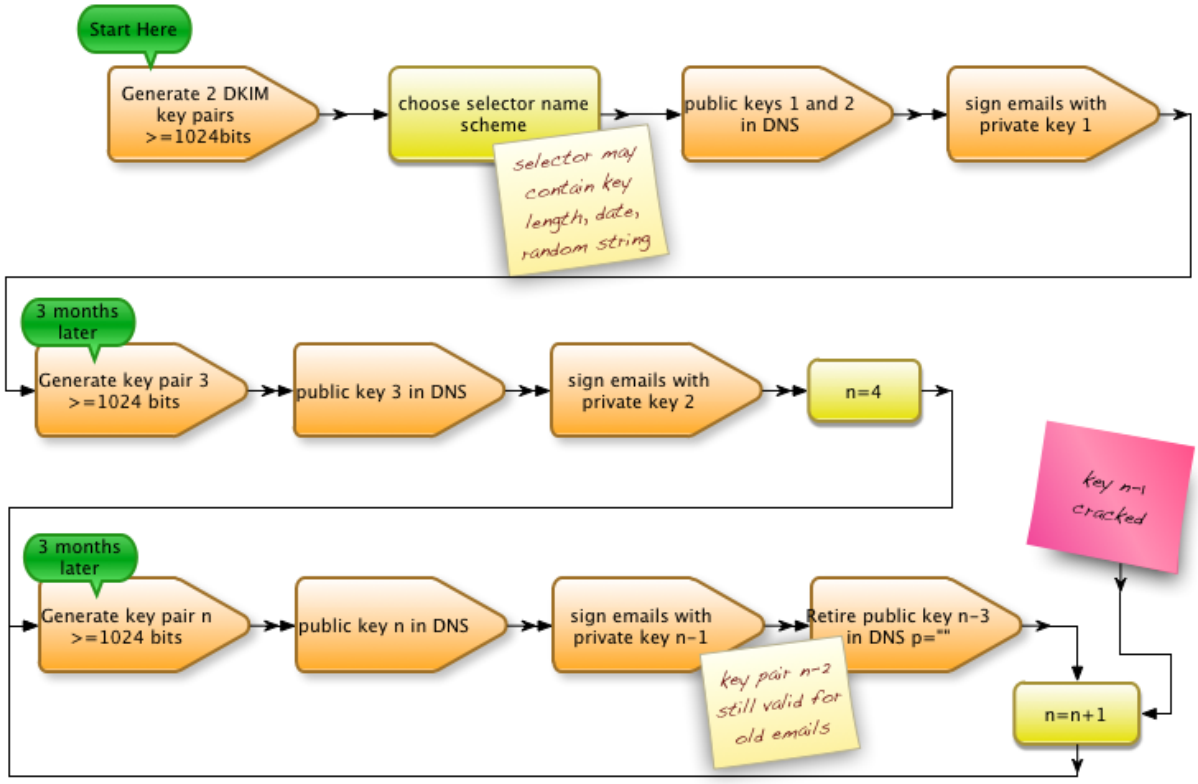
4.5 Disaster Happens, Be Prepared

At any time, one of your key pairs can be compromised so be prepared to require the appropriate parties to initiate an unscheduled, emergency key rotation. It is best if there is significant preparation to quickly accomplish this, especially by pre-publishing the next public key in the DNS. In this way, emergency efforts will only required by the mail administrator, which represents the least number of people that needs to be involved in the process.

5. Specific Steps of Rotation

5.1 Timetable and Workflow Process

This is a proposed workflow that should suit most needs and can be modified as appropriate to varying situations. It is important to document each step of the process to ensure you capture how your organization specifically handles the rotation. The workflow illustrates a quarterly rotation but the same timeline can be changed to six months for a bi-annual rotation.



5.1.1 Define Your Selector Rotation Scheme

The selector points the email receiver to the public key in your DNS. This selector should carry enough information to facilitate the key rotation process and be significantly complex to prevent the next public key from being easily guessed, if it is published in the DNS.

NOTE: *See Section 4.3 for guidance on defining an effective naming scheme.*

5.1.2 Getting Started

If this is the first time you will use DKIM for your email, generate two key pairs. If you are starting the rotation process and already have a DKIM key pair in place, start by generating another key pair. In either case, publish the public key(s) in your DNS.

Before signing any email, wait for the DNS modification(s) to propagate. Once you have confirmed that the keys have propagated, start signing the email you send.

If you are already signing your email with a previous key, wait for the new keys to propagate through DNS before continuing.

At this step you have two key pairs generated and the associated two public keys in your DNS. Key pair 1 is used to sign email, key pair 2 is ready to be used.

Example:

k20130103-1024-dfrstu._domainkey.example.com <-- public key 1 used to sign email
k20130409-1024-htedrp._domainkey.example.com <-- public key 2 for future use

5.1.3 The First Rotation

Generate a new key pair (key pair 3) and publish the public key in the DNS.

Stop signing email with your first key pair and start signing with your second key pair. At this step, you have three valid key pairs and their associated public keys in the DNS.

Key pair 1 is still valid allowing receivers to still validate old email or email that was still in transit while you changed the signing to key pair 2. Key pair 3 is ready for future use.

Example:

k20130103-1024-dFrTu._domainkey.example.com <-- public key 1 to validate old email
k20130409-1024-htdRp._domainkey.example.com <-- public key 2 in use to sign email
k20130709-1024-hxdVb._domainkey.example.com <-- public key 3 for future use

5.1.4 Subsequent Rotations

Generate another key pair (n) and publish the public key in the DNS. This key pair is for future use.

Stop signing email with the current key pair (n-2) and start signing with the next key pair (n-1).

Retire the key pair (n-3) that had still the public key in the DNS by setting the "p" field to be empty (". . . p=").

NOTE: *Do not delete the key entry from the DNS as it should still be discoverable, the empty "p" field indicates that the key was knowingly retired.*

Example:

k20130103-1024-dfrstu._domainkey.example.com <-- public key 1 is retired p=""
k20130409-1024-hterp._domainkey.example.com <-- public key 2 validate old email
k20130709-1024-hxdvb._domainkey.example.com <-- public key 3 used to sign email
k20131004-1024-hrtdgb._domainkey.example.com <-- public key 4 for future use

6. Auditing

Based on your organization's security policy, you should perform regular audits of your DKIM key rotation. A good practice would be to schedule audits shortly following a rotation (e.g., one week later) to ensure the rotation took place and was successful. Start collecting all necessary information in a data store as soon as you establish the key rotation process. For each mail stream in your organization, track all domains (and subdomains) and all DKIM selectors with their activity time ranges. It is also helpful to note the key personnel responsible for each mail stream.

To perform an audit, compare the current domain ("d=" value in the DKIM signature) and selector ("s=") combination for a particular message with the expected values found in your DKIM rotation data store. Having a strict selector naming scheme will streamline ad-hoc audits and can be the basis for automated auditing.

For example, assume that you are auditing after a rotation planned for the first week of April and your selector follows a naming convention that includes the date when the key pair was planned to go into active use. In this case, the active selector should include the date string such as "20130401" (i.e., April 1, 2013). Thus, all audited messages should indicate that they are signed using the correct selector.

If, however, audited messages indicate that they were signed using another selector, this may indicate a problem with the rotation. Further, depending on your naming scheme, the selector name should provide information to troubleshoot the difficulty.

References

Relevant IETF RFCs:

1. RFC6376 - DomainKeys Identified Mail (DKIM) Signatures - http://tools.ietf.org/html/rfc6376
2. RFC5585 - DomainKeys Identified Mail (DKIM) Service Overview - http://tools.ietf.org/html/rfc5585
3. RFC5863 - DomainKeys Identified Mail (DKIM) Development, Deployment, and Operations - http://tools.ietf.org/html/rfc5863

References on Key Lengths:

1. http://tools.ietf.org/html/rfc3766
2. http://www.keylength.com/
3. M³AAWG Best Practices for Implementing DKIM To Avoid Key Length Vulnerability (http://www.maawg.org/sites/maawg/files/news/M3AAWG_Key_Implementation_BP-2012-11.pdf)

Other Articles

1. "How a Google Headhunter's E-Mail Unraveled a Massive Net Security Hole" by Kim Zetter, Wired, October 24, 2012 (http://www.wired.com/threatlevel/2012/10/dkim-vulnerability-widespread/).
2. ". . .recommend[s] that keys be rotated quarterly, and that signatures should have an expiration period greater than the current key rotation period." (http://www.scmagazineuk.com/experts-publish-dkim-flaw-best-practice/article/267667)
3. ". . .keys may be rotated relatively easily. Domains that want to use 512-bit keys should rotate their keys relatively often." (http://blogs.cisco.com/security/key_lengths_for_dkim_signatures/#more)
4. "INFORMATIVE OPERATIONS ADVICE: A signer should not sign with a private key when the selector containing the corresponding public key is expected to be revoked or removed before the verifier has an opportunity to validate the signature. The signer should anticipate that verifiers may choose to defer validation, perhaps until the message is actually read by the final recipient. In particular, when rotating to a new key pair, signing should immediately commence with the new private key and the old public key should be retained for a reasonable validation interval before being removed from the key server." (http://dkim.org/specs/rfc4871-dkimbase.html)