Using the KEMP LoadMaster™

Moving to 2048 SSL Keys and the Impact on Computing Resources

Jon Malek, Senior Systems Test Engineer,
KEMP Technologies

Moving to 2048
SSL keys and the impact on computing resources

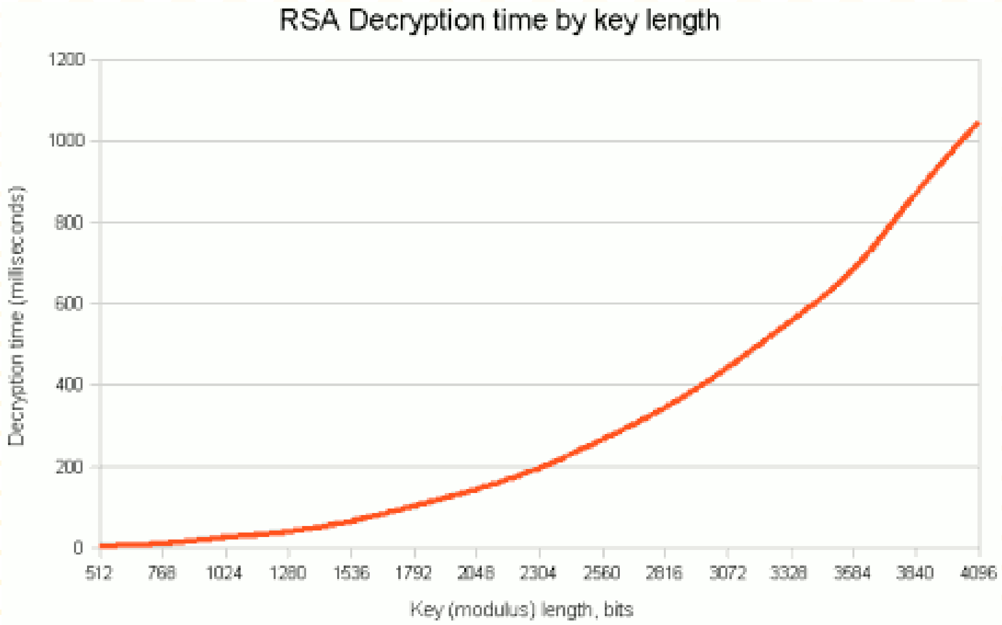
Why should you move to 2048 bit keys?

In order to keep your connection secure, SSL/TLS employs a mechanism to ensure that the server you are connecting to is indeed the server it is claiming to be, and that no one can read your data. It does this by sending a certificate which has been vouched for by trusted certificate authorities (CA). The server then uses session data provided by both the client and the server and private key to encrypt and decrypt the traffic during the session. The length of the server's private key is responsible for how difficult it is to crack the encryption.

Up until recently, a 1024 bit RSA key length was considered strong enough for every day encryption. However, as computers have gotten stronger, this is no longer considered the case. Key lengths less than 2048 bits are now considered deprecated by the NIST (Barker, Elaine; Roginsky, Allen;, 2011). It is therefore imperative that websites move to at least 2048 bit key lengths to ensure a secure connection. Indeed, most CA's will only issue 2048 bit certificates.

What is the impact on server resources?

Most of the performance cost during an SSL/TLS session occurs at the beginning, when handshaking; the longer the key, the more CPU intensive the handshake. In fact, the increase in key length corresponds to an exponential increase in CPU usage. As compared to 1024 bit key lengths, 2048 bit key lengths can take upwards of 4 times as much CPU usage. The effect is more noticeable in sessions where the pages themselves are small, and sessions are short. This usually means that there will be more new connections, and therefore, more SSL/TLS handshakes squeezed into shorter intervals. This will almost always translate to using up most of your CPU resources on dealing with SSL/TLS handshaking. On a busy server, this could cause noticeable delay when connecting to the site, especially while under heavy load.



As a consequence, increasing the key length from 1024 to 2048 reduces how many new connections per second a server can handle by about 75% to 80%. This is a dramatic drop in performance, which could severely cripple an underpowered server.

What can be done to alleviate this performance drop?

Depending on the server, they may be able to be upgraded to include a more powerful processor. It may also be possible to add in a cryptography ASIC card. Depending on the system in question and budget, this may or may not be possible. Another option would be to place an SSL accelerator appliance in front for your site(s). SSL accelerators are also typically found inside Application Delivery Controllers (ADC) and Hardware Load Balancers. Any of these devices could offload the SSL/TLS from the server, providing more specialized hardware to handle the encryption.

Conclusion

While necessary for maintaining strong security for a site, 2048 bit RSA key lengths are very processor intensive; quite a bit more (upwards of 4 times) intensive as 1024 bit keys. Before moving to these key lengths, it is important to understand the effect on the system it will have.

Works Cited

Barker, Elaine; Roginsky, Allen;. (2011).
Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. Computer Security Division, Information Technology Laboratory, NIST.

Coffey, N. (2011).
RSA Key Lengths. Retrieved March 24, 2011, from Javamex Home: http://www.javamex.com/tutorials/cryptography/rsa_key_length.shtml