SUMMARY

Five years ago, widespread tools for strong encryption were uncommon. Back then there were all sorts of crypto and the weak ones have been diminished from the crypto pool. Now we have all the best crypto available and this makes the new generation of crypto troublesome. New crypto software today is developed under the assumption that the developers are experienced in their field and the crypto knowledge required for safe use. The modern developer of crypto software is unmotivated in the sense that they only do what their boss wants.

There is very little research on using crypto in a secure fashion. This is mainly because authors present the algorithms and mechanisms to the developer and the developers must figure out what to do with it. Another reason why there is very little research on using crypto in a secure fashion is that developers patch up programs when they find the problem and not thinking about it beforehand.

Crypto and security software can be misused in many ways. There is no universal fix for Crypto problems, but there are some guidelines that will give developers exposure to the existence of these problems. For example, Cryptlib has a major design feature which is that it doesn’t expose private keys to outside access. Companies would spread private keys around because it’s cheaper than purchasing a certificate of the same amount to spread the key safely. A solution to managing keys is if XML is used the key is put in the field labeled key or password. Another solution to key management is to use a fixed key throughout an organization by using WEP.

There are security protocols put in place called PKI protocols that dealt with time. This pose a problem when user computers would have the wrong time on their machine. This is a problem for time-based certificates as it used the time that’s on the user’s computer. A solution would be to use a nonce.

When doing bulk encryption, people use pure RSA instead of using it as a key exchange. When a customer wants an RSA encryption of data, developers implement it the way the API wants it to be.

Users are left with an exercise by the crypto tools used by the toolkit developers. An example would be the entropy data for key generation. This is expected of the user and experienced users do everything they can to input entropy to a random number generator.

A problem occurred when using RSA to wrap a symmetric encryption key and used to encrypt messages being sent. This technique resulted in the format of the message being correct, however a tester realized that some parameters of the key had a slight error. The RSA encryption code was receiving the wrong parameters, but the format was correct. The function that passed the parameters to the RSA encryption was failing with an error code, however this function couldn’t fail so the programmer didn’t check the parameters that was being passed. When the message was sent, the receiving end had the same problem. Everything looked fine, but symmetric key was not used by the RSA decryption code. To avoid this type of problem, the design of the function should make it fail safe.

Unexperienced users are mostly average people that has had some knowledge to cryptography but not actual training. These users are the ones misusing crypto by creating snake oil and Naugahyde crypto. The solution is to have the libraries be at the highest level possible so that low level users can’t hurt themselves.