# Defeating SkyNet Part 2: Design Document

*Stephen Tridgell – 309205867 , Nicholas Carydis – 310183626*

To properly defend a botnet from governments and other hackers, security features must be implemented. One important area of security involves the integrity and confidentiality of the data flowing between the bots and the master. Part 2 of “Defeating Skynet” entailed the implementation of a security protocol to ensure the integrity and authenticity of botnet updates, as well as asymmetric encryption to protect the confidentiality of the sensitive data extorted by the bots.

* **How do you ensure the only one who can send updates to SkyNet is the botnet master?**

A digital signature scheme was chosen to ensure the authenticity of the updates sent to the bots. A module was developed to allow the master to sign any file present on the pastebot.net website. This was achieved using an asymmetric cryptography system (the PKCS1\_PSS module within the PyCrypto library), as well as the SHA hash function. On the master-side of the operation, a file would first be selected from the pastebot.net folder. This file would then be signed by hashing it, signing the hash and appending the signed hash to the original file. The signature was done using the master’s 4096 bit private key. The filename was then appended with ‘.signed’ and uploaded to the pastebot site for distribution.

The bot code was then modified to ensure that any downloaded file (whether it be downloaded from pastebot or from a fellow bot) was signed by the master. This was done by first decrypting the 512 B of the file (the signature) using the master’s public key (which is known to all bots). The resulting value was the proclaimed hash value of the file. The rest of the file would then be hashed by the bot using the same SHA hash function, and the two hashes compared. If the comparison fails, the file would be neither executed nor stored by the bot.

* **How do you protect the valuable information to ensure it can only be read by the botnet master?**

As the valuable information extorted by the botnet is stored on a publically-accessible pastebot server, it would be prudent to ensure its confidentiality in some way. The requirement in this case was for the master to be the only recipient authorised to read the files. This was met using asymmetric encryption once more.

Any time a bot is to upload valuables to pastebot, those valuables are encrypted using the PKCS1\_OAEP module of the PyCrypto library. Due to the maximum block size being (RSA Modulus – 2 – 2x Hash length), the maximum block size used by the bot implementation was 470 bytes. This produced a cyphertext of 512 Bytes. To accommodate files larger than 470B, the bots would split the files into blocks, encrypt each block and concatenate all encrypted blocks together before uploading. The encryption was performed using the master’s public key, ensuring the valuables can only be read by the master.

The master could then read an encrypted file by splitting the cyphertext into blocks of 512 Bytes, and decrypting each one with its private key.

* **How do you ensure the botnet updates signed by the botnet master cannot be forged or modified?**

The digital signature algorithm applied to botnet updates by the master ensures data integrity along with authenticity. The digital signature is hard to forge, due to the large key used to sign the hash needed to generate the signature. To achieve a forgery would require an adversary to break the asymmetric encryption involved in the signature by obtaining φ(n), which has been widely accepted to be a difficult task.

The update data is also hard to modify, due to the second pre-image resistance property of the SHA hash function used. To achieve a modified update would require an adversary to find a different file that hashes to the same value as the authentic update file, believed to be a difficult task due to the reason above. Of course, such a file would probably be malformed and unexecutable, and thus such an attack would at most prevent a new update from being run rather than change anything of value. Thus if the goal is to spread a useful unauthorised update, an attacker would be better served by attacking the encryption component of the digital signature.

* **If SkyNet’s botnet code is dismantled and/or the source for it stolen, does your scheme become less secure?**

Knowledge of the schemes used to ensure integrity and authenticity of updates and confidentiality of valuable information does not significantly reduce the security provided by implementing these schemes. This is due to the fact that the security rests on the strength and privacy of the master’s private key. The scheme was developed without any assumption of implementation obscurity.

* **Give an indication of how difficult it would be for an adversary to take control of SkyNet when your protections are used.**

The possibility of a total takeover of SkyNet by an adversary only exists within the limits of a private-key compromisation. Thus the security of the master’s private key is the deciding factor when analysing SkyNet’s susceptibility to a takeover. It is therefore important that the master keep the key secured (perhaps by encrypting the key using a known password), the location of the key secured (so as not to be targeted by hackers or agents remotely), and his/her identity protected (so as to avoid the use of rubber-hose cryptography).