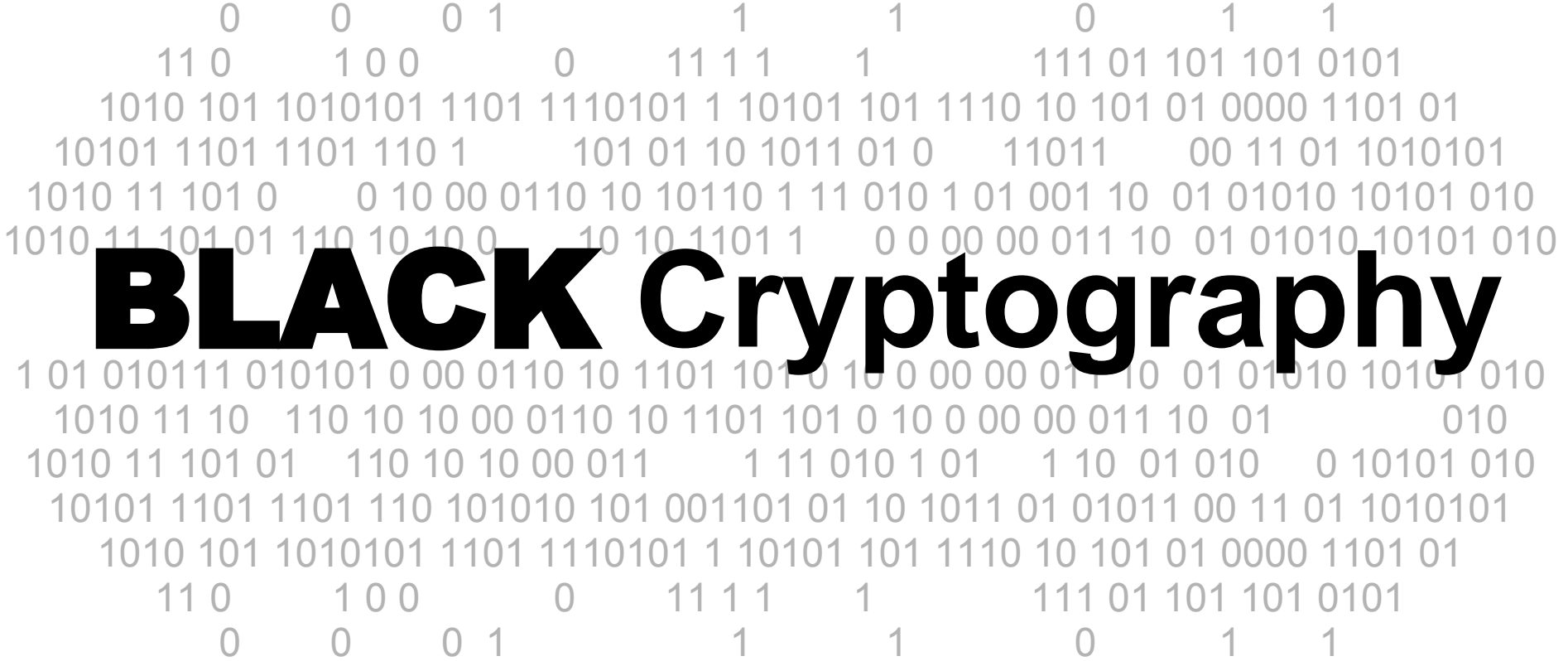

BLACK Cryptography

What is BLACK Cryptography?

Cryptography has more applications than just protecting your credit card numbers. Beyond the convoluted and often 'boring' math, there are numerous hostile applications for you to take advantage of:

Hard Drive Extortion

Polymorphic Viruses

Botnet Herding

Hash Collisions

A *Very* Brief Overview

What you'll need to know for this talk. If these seem foreign, this bit is for you. If you already know these, take a brief nap.

- Public vs. Private Key Encryption
- Private Key Algorithms
  - AES, 3DES, DES
- Public Key Algorithms
  - RSA, Diffie-Hellman
- Hash Algorithsm
  - SHA-2, SHA-1, MD5

Public vs. Private Key

Private Key:

- Single key shared between two parties.
- Security relies on the length of the key and obfuscation.
- Very quick
- Hard to “prove” security of the system – run it through convoluted tests, if it passes, it's secure.

Public Key:

- Two key system. One is made available to everyone, one is kept secret.
- Security relies on “one-way” math functions.
- Relatively slow.
- Security proofs are based on computational difficulty.

Private Key Algorithms

Private key encryption is used to transfer messages exclusively between two parties. Revealing the key to anyone else would allow them to decrypt messages.

- The primary difficulty of private key encryption is sharing the key before encryption.
- Once a shared key is generated, encryption and decryption follow a simple protocol (AES or DES):
  - $Dec_K(Enc_K(M)) = M$
- One other difficulty with private key encryption is there is no ability to determine the sender of message. This excludes the possibility of digital signatures.

Public Key Algorithms

The goal of public key encryption is to share secrets with multiple parties without ever establishing a shared secret key.

- Generate a key pair (PK, SK) using RSA or Diffie-Hellman.
  - PK, your public key, is placed in the open for anyone to use. For most of our cases, this will actually be saved in malicious code.
  - SK, your secret key, is stored within your own machine or smart-card.

Public Key Algorithms, continued...

Once you have your (PK, SK) pair, you can do two types of operations:

- Confidentiality: Encrypt messages with a party's PK, only that party can decrypt with the SK.
  - $Dec_{PK}(Enc_{SK}(M)) = M$
- Authentication: Encrypt a message with your own SK, any party can verify you're the message sender by decrypting with your PK.
  - $Dec_{SK}(Enc_{PK}(M)) = M$
    - Also known as a digital signature.

Hash Algorithms

The purpose of a hash is to create a simplified view of a file that detects tampering or modification.

- Example: Take an ISO of an operating system, place it through a hash algorithm, post the hash output on a website.
  - Users then download the ISO via torrent files or mirrored websites, checking the hash value of the downloaded file against the posted hash.
    - If they match, the user 'knows' the file hasn't been modified.
- MD5, SHA-1, SHA-2 are all examples of hashes.
  - More about weak hashes and hash collisions later.

Hard Drive Extortion

Hard Drive Extortion

The strength of encryption is a double-edged sword. When you encrypt your hard drive to prevent unwarranted access, reading becomes limited only to those who have the key.

- Now reverse the roles. Rather than keeping malicious users from reading your HD, malicious users can take your HD hostage and prevent you access.

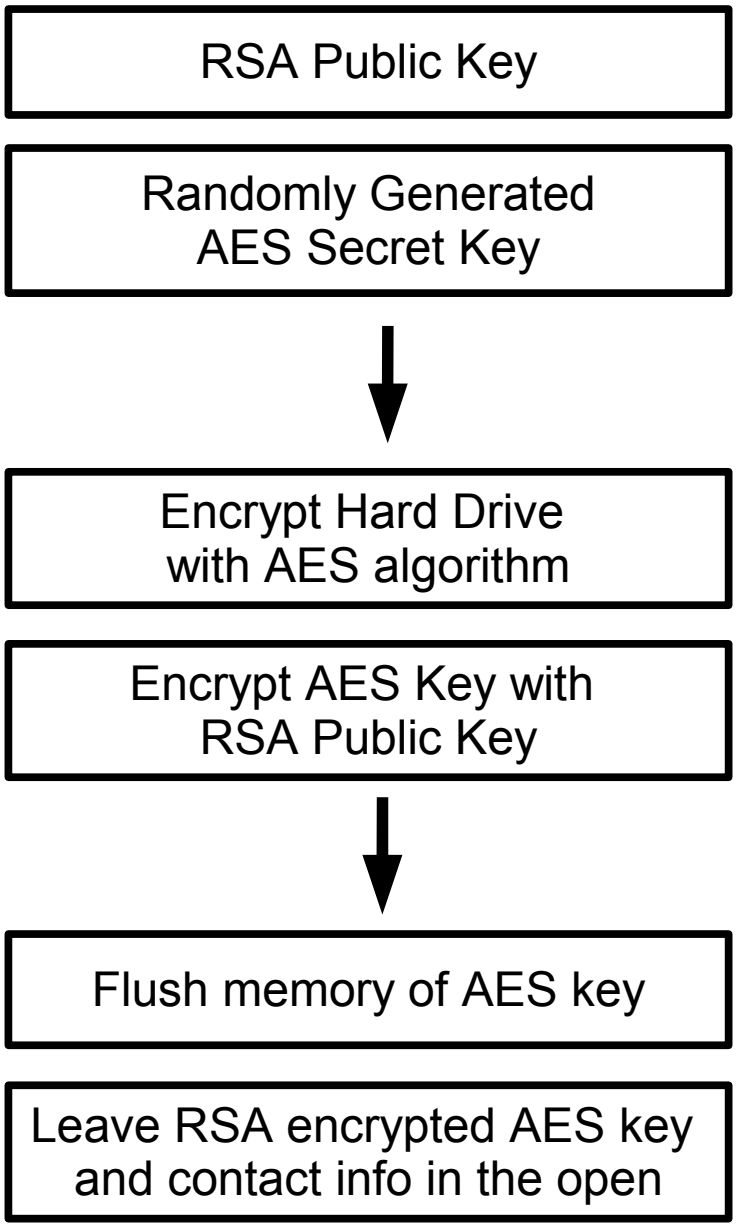
Enter hard drive extortion...

Hard Drive Extortion, continued

Consider a situation:

- A virus infects a machine and encrypts its hard drive against the users will.
- Without knowledge of the key, strong encryption will be impossible to break.
- Hard drive can either be tossed out, or extortion can be paid.

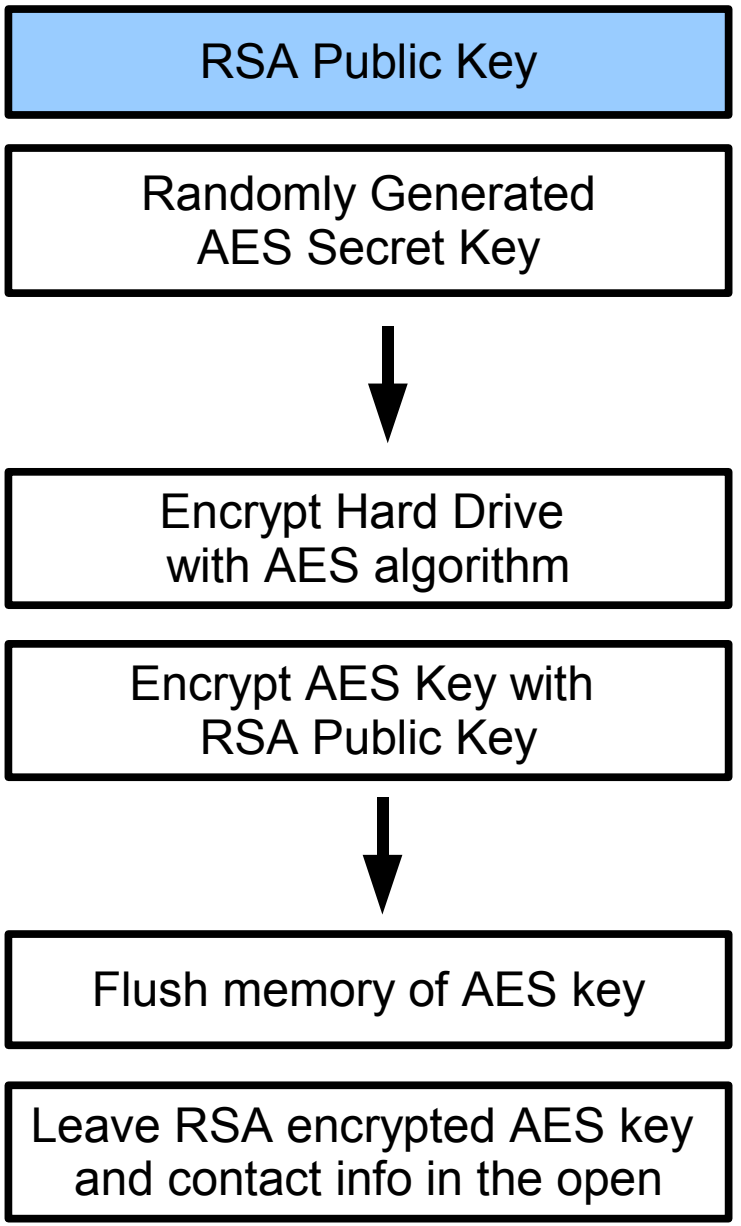
Here's how it works ...



Hard Drive Extortion, continued

Breaking it down:

When Eris first writes her virus, she generates a key pair (PK,SK) and stores the PK within the payload of the virus.

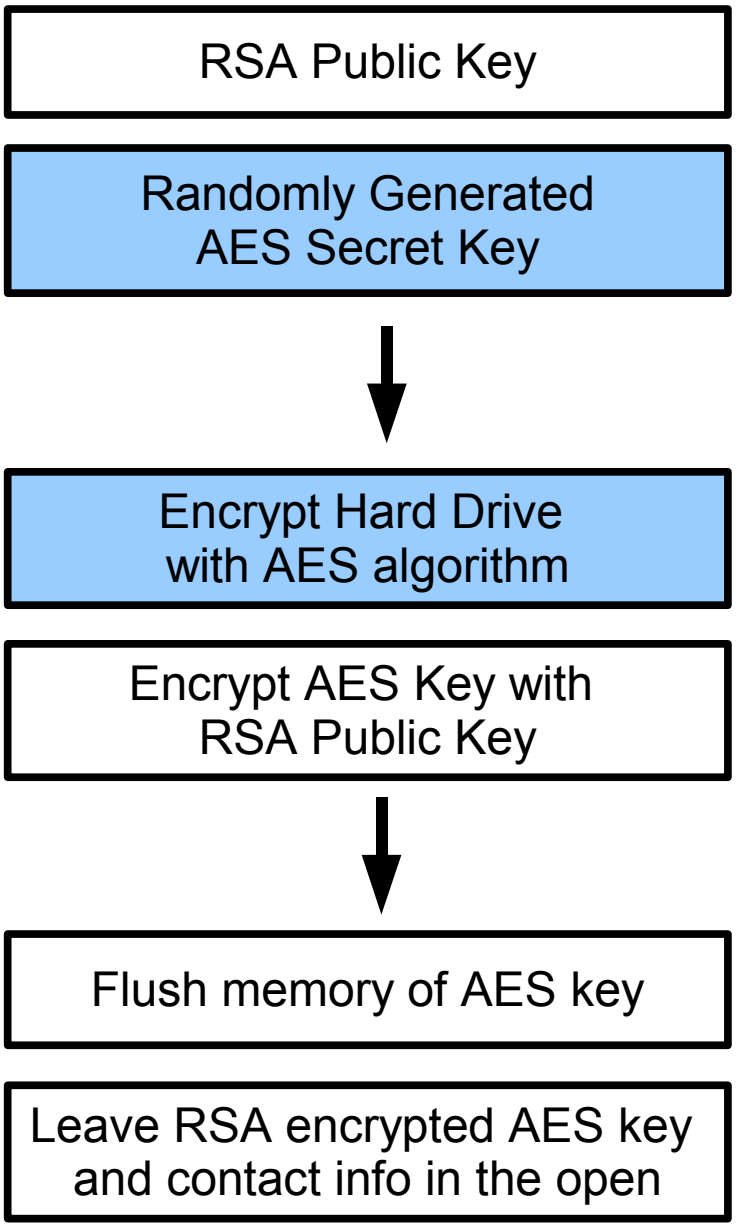


Hard Drive Extortion, continued

Encrypt the Hard Drive:

The next step is to randomly generate a key for a Private Key encryption scheme, in our case, AES.

That key is then used to encrypt the contents of the hard drive.

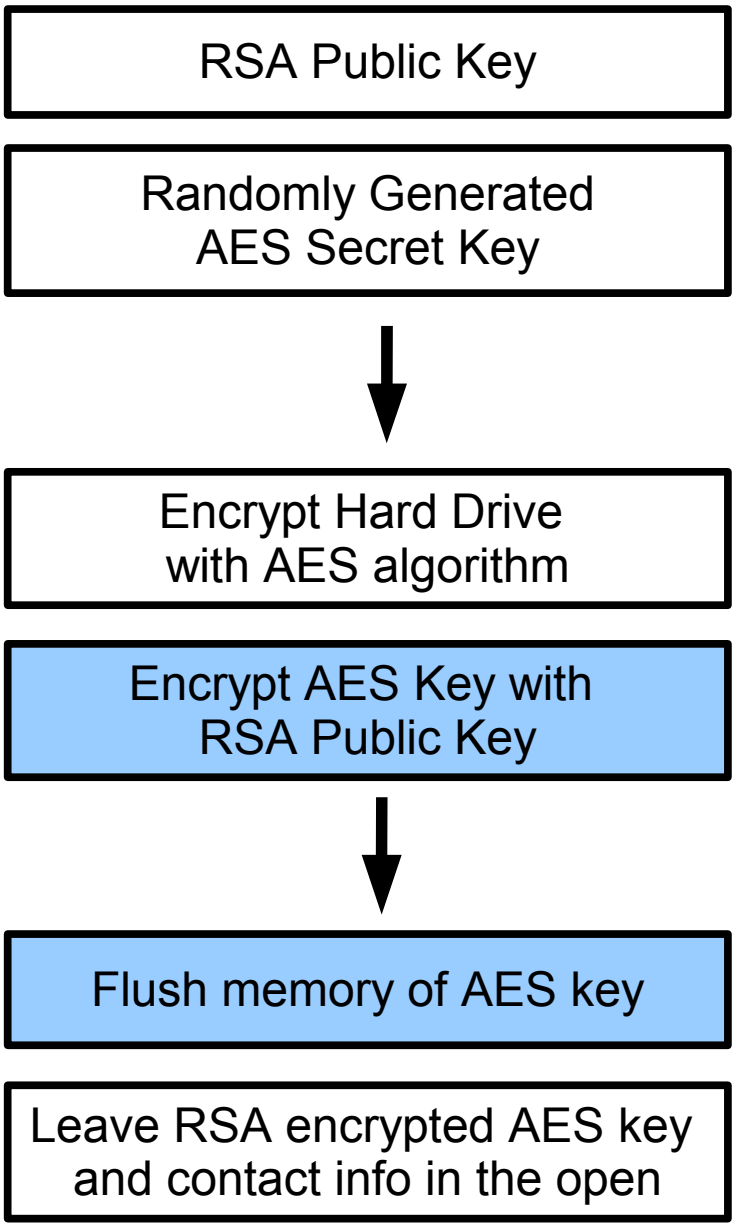


Hard Drive Extortion, continued

Save the AES key for decryption at a later time.

By encrypting the AES key with the RSA public key, only the Eris now knows how to decrypt the HD.

After flushing out the memory of the AES key, she's on her way to extortion.



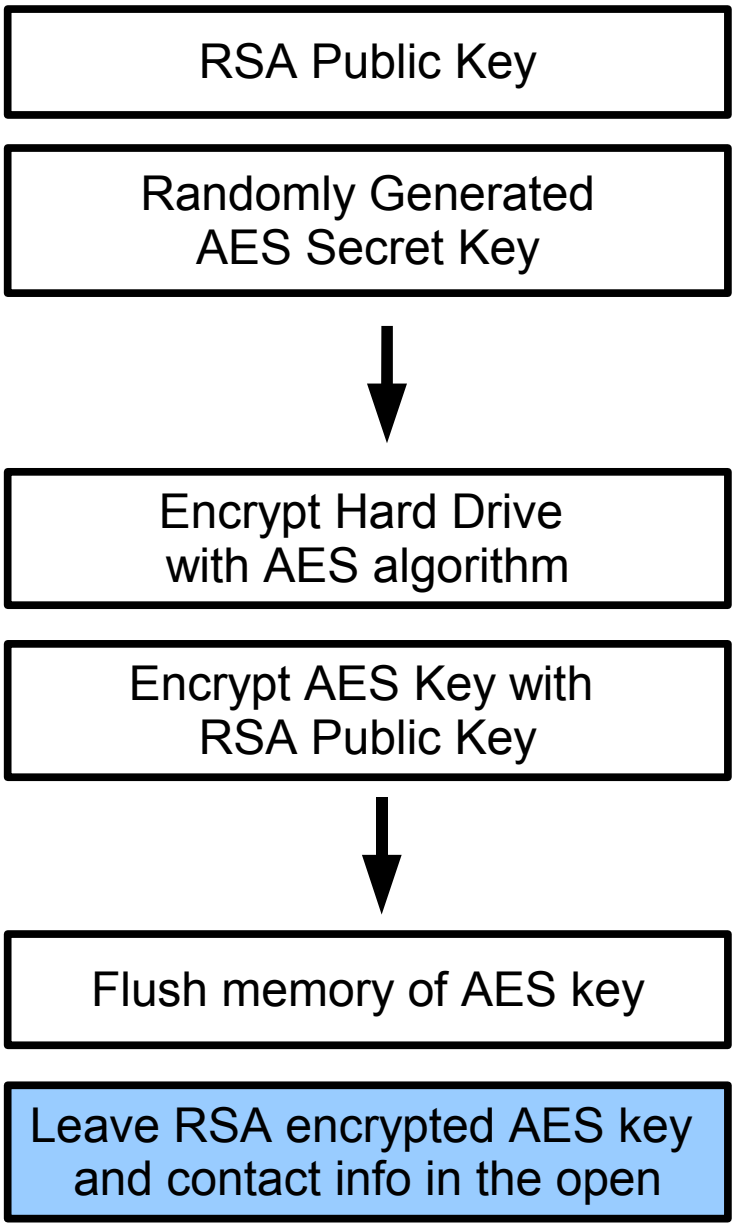
Hard Drive Extortion, continued

(1) Encrypt a hard drive

(2) ???

(3) Profit

- Eris has one problem to overcome, getting the money to her account. Time to contact the Swiss, the Caymans, or E-gold.
- Once ransom has been paid, Eris decrypts the signed AES key and reveals it to the client.



Hard Drive Extortion, finalized

Why the complication?

- By combining both public and private encryption algorithms, Eris achieves a virus that scales to as many machines as she can infect.
- Because the AES key generated each time is random, revealing the key of one infected machine reveals nothing about a second machine.

What's more difficult is remaining anonymous throughout the extortion. Tor or any other mix network can hide Eris' identity, but bank accounts may be her undoing.

Polymorphic Viruses

Polymorphic Viruses

One of the problems of viruses is their detectability because of static structure.

- Once a virus is in the wild, anti-virus companies design methods for detecting the virus' code signature.
- Whenever similar code appears, a flag is waved.

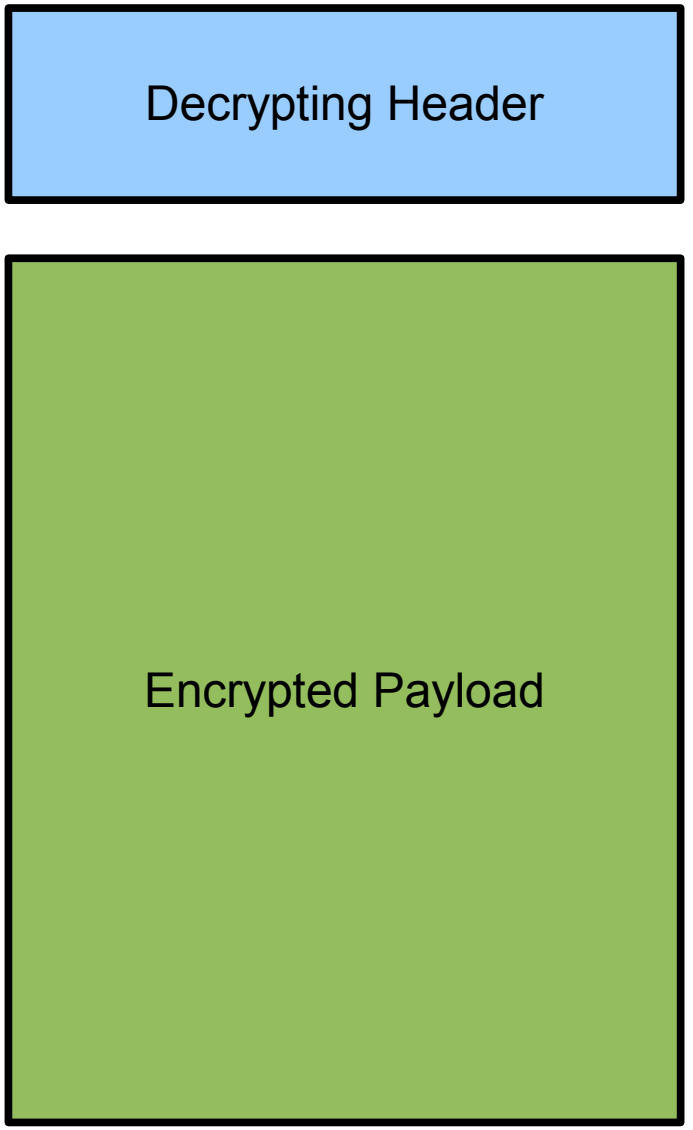
The task for virus writers is to design viruses that modify their own code and behavior making it difficult to generate signatures to identify the virus.

- Two methods:
  - Obfuscation through encryption.
  - Obfuscation through randomized, but meaningful instructions and structure.

Polymorphic Viruses, continued

Rather than revealing the payload of a virus to random scanning, the majority of code is obfuscated through encryption.

- A small header file contains a randomized key and a tiny algorithm to decrypt the payload.
- With each propagation, a new key is generated and used to encrypt the payload.



Polymorphic Viruses, finalized

Detection must rely on an increasingly smaller sample size for just random scans.

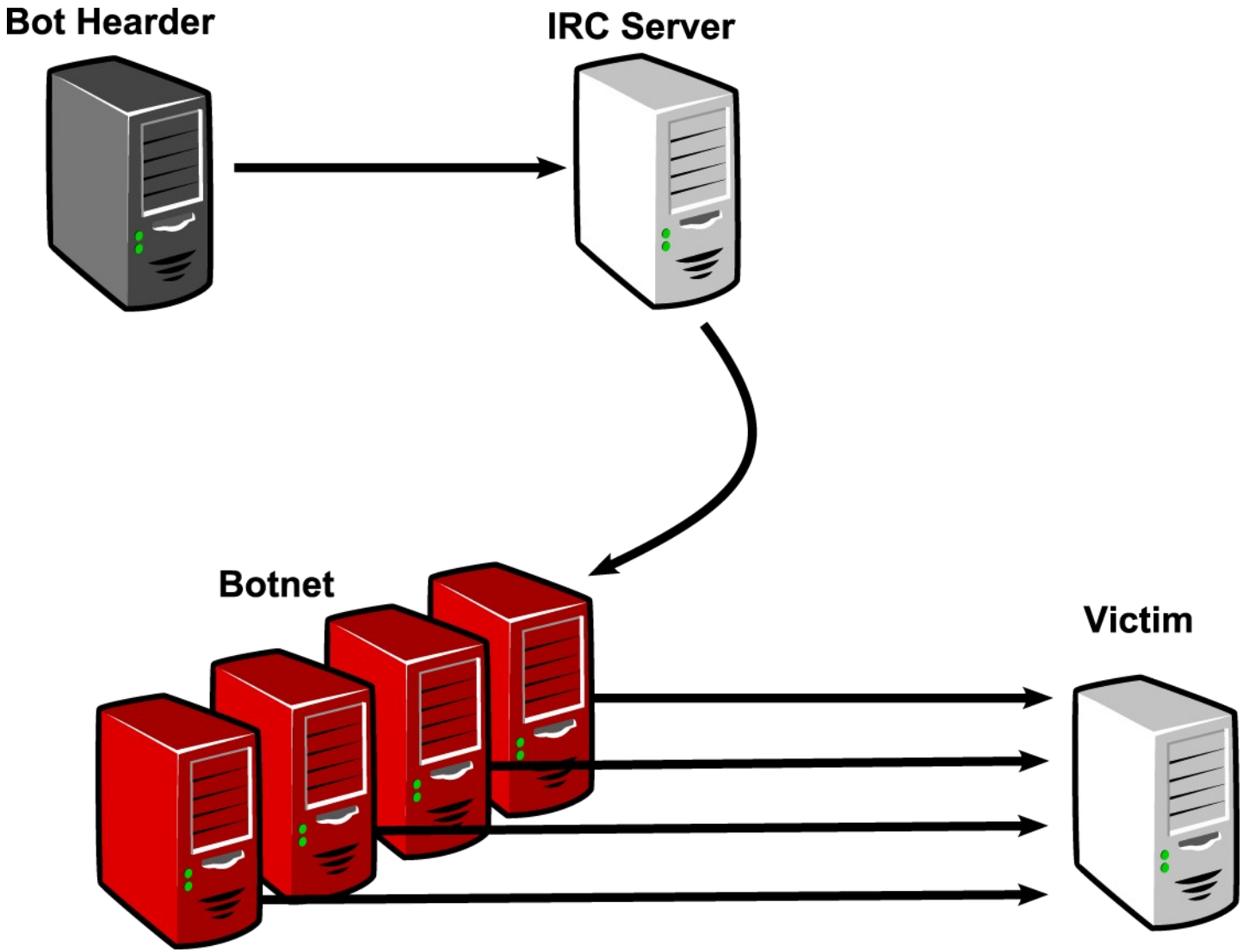
- A virus scan can attempt to recognize the virus by its attempt to use encryption. Many programs use encryption though...
- Additionally, it can try to fingerprint the header file.

What responses by the writer?

- Obfuscated and randomized instructions.
  - Add in NOPs, unused ADD, XOR, and JMP instructions
- Multiple smaller headers.
  - Randomize the structure to make fingerprinting difficult.

Botnet Herding

Anatomy of a Botnet

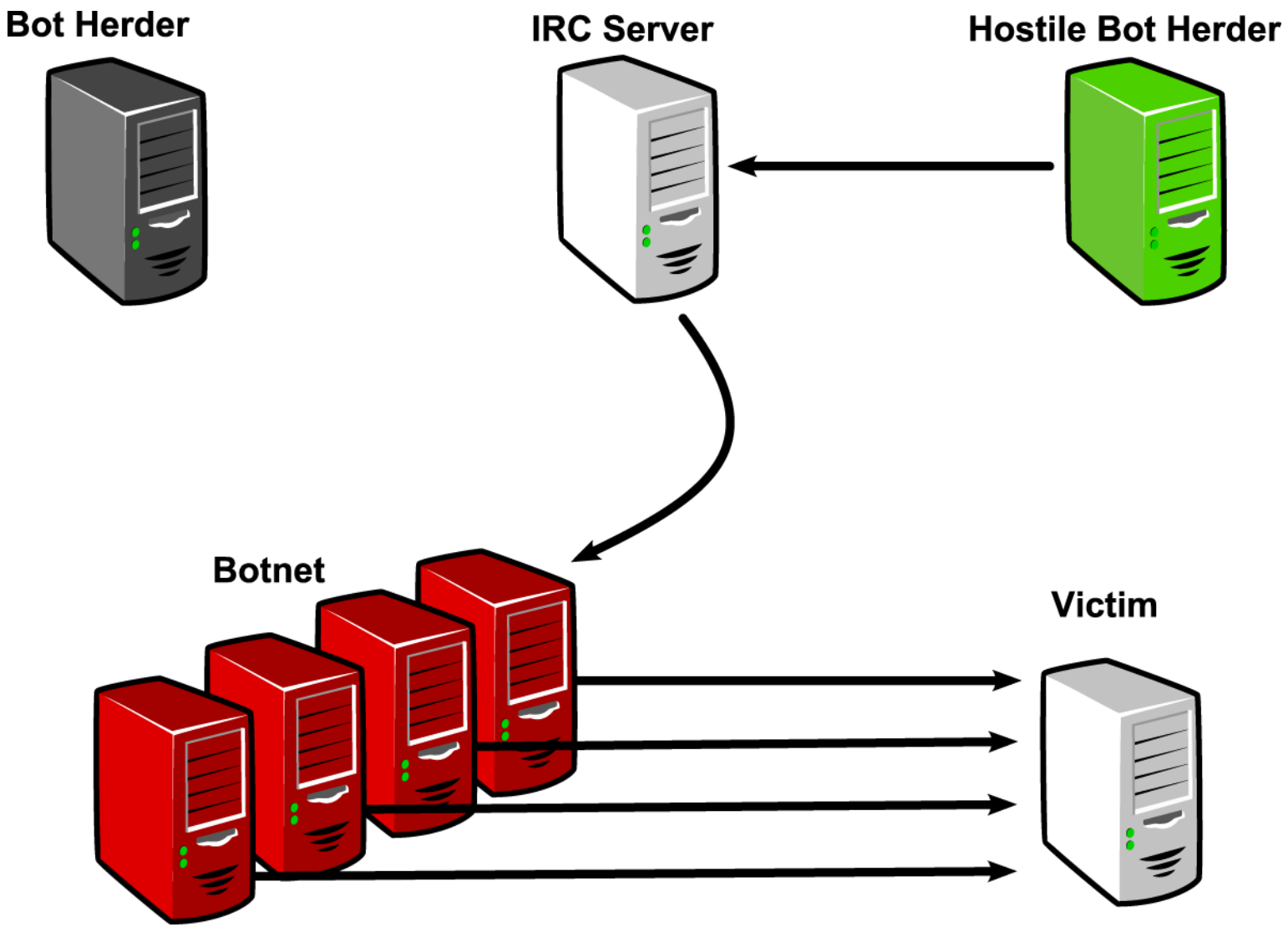


Botnet Hearding

So what are the problems?

- If a bot virus ever infects a honeypot or hostile warden machine, it reveals the IRC channel and password necessary to connect.
- Once a hostile entity logs into the Bot herders IRC channel, opens the possibility for directing the network.
- Additionally, if a bot machine in the wild is identified, a hostile herder can either:
  - Try and monitor traffic going to and from the bot from a network standpoint.
  - Break into the machine and monitor traffic in an attempt to locate the host IRC channel.

The Problem, Illustrated



Botnet Herding, finalized

To prevent another entity from ever directing the network:

- When the bot program is installed, the bot herder can include their PK from a (PK,SK) pair.
- All instructions generated by the herder are hashed and signed with the herder's SK.
- A random number is included to prevent a hostile herder from replaying attack.

Inst||Rand Number||Enc_{SK}(Hash(Inst)||Rand Number))

- If the network is being loaned out for money, the owner can just sign instructions generated by the client.

Hash Collisions

Hash Collisions

Hashes provide a simple way to verify the contents of a file, but what if two files hash to the same value?

- Hash collisions are supposed to be infrequent, 1/2^80 for a 160-bit hash and a 'random' distribution.
  - Reality seems to show otherwise, MD5 is thoroughly broken and SHA-1 has numerous papers about collisions.
- What can be done with collisions?
  - Hide malicious code inside trusted entities.
    - Any open source program that is distributed over a torrent can become a method of distributing malicious code.
    - No more need for buffer overflows, abuse people's trust instead.

More Than Just Infancy

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

To Whom it May Concern:

Alice Falbala fulfilled all the requirements of the Roman Empire intern position. She was excellent at translating roman into her gaul native language, learned very rapidly, and worked with considerable independence and confidence.

Her basic work habits such as punctuality, interpersonal deportment, communication skills, and completing assigned and self-determined goals were all excellent.

I recommend Alice for challenging positions in which creativity, reliability, and language skills are required.

I highly recommend hiring her. If you'd like to discuss her attributes in more detail, please don't hesitate to contact me.

Sincerely,

Julius Caesar

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

Order:

Alice Falbala is given full access to all confidential and secret information about GAUL.

Sincerely,

Julius Caesar

Hash Collisions, finalized

Hash functions currently just obfuscate data with the hope that somehow the end result is impossible to reverse or easily reproduce.

- MD5 has been thoroughly analyzed and manipulated.
- Collisions have also been produced for SHA-1

The NIST is already in the process of designing an Advanced Hash Standard (AHS).

- In the mean time, the NSA recommends using SHA-2 variants: SHA-256 and SHA-512.

Resources

- Introduction to Cryptography with Coding Theory
  - Wade Trappe & Lawrence Washington
- Applied Cryptography
  - Bruce Schneier
- Handbook of Applied Cryptography
  - Free online
- Malicious Cryptography
  - Adam Young & Moti Yung

Courses at UIUC

For those interested in cryptography and not algorithms:

- Math 453, Elementary Number Theory
- Math 417, Introduction to Abstract Algebra
- Math 5**/ECE 559, Coding and Cryptography
  - They change the number every year...
- CS498, Theoretical Foundations of Cryptography
  - A bit of a snore, but what's covered is used in real theory