COIS 4310H – Assignment #3
Simon Willshire (0491272)

1. Should node 18 join, finger tables need to display the correct successor, so that the highlighted rows have been changed:

Node 1

2	4
3	4
5	7
9	12
17	18

Node 4

5	7
6	7
8	12
12	12
18	18

Node 12

13	15
14	15
16	18
18	18
20	20

Through the addition of node 18, the finger tables need to shift the use of node 20 to 18, and readjust the successors accordingly (see above).

2. The purpose of a good content delivery network is to be readily available and high performance for live streaming data. For a single ISP to work as a CDN, the ISP would need to adapt their model of delivery substantially. Typical CDN's consist of mirrored sub-networks that act to distribute the load in the area, that is to say that typical CDNs are made up in multiple geographical areas. In order for an ISP to operate as a CDN, they would need to be sufficiently big enough geographically (for example Nexicom would be a poor choice as a local provider, but an ISP with international support would be more acceptable). As a whole the typical ISP configuration would make for a poor infrastructure to run a CDN on.

3. Attack on DES
- Assume plaintext ASCII values{65-90, 44, 46, 59, 13, 10}, or 30 unique char
- Nothing known about parity bits
- Assume NSA-tampered 56bit key (7 bytes/chars, but uses 8 byte chunks programmatically)

Brute Force: 2^{56}key space

Plain-text assumptions:
The output of the decryption needs to fit within the 30 unique character sequences, so we can reduce the search space from 255 combinations to 30 possible byte values, giving us $\left(\frac{\mathbf{30}}{\mathbf{255}}\right)^{\mathbf{8}} = \mathbf{4}$ blocks to decrypt per key check/validation, from an original 8, reduction in half of the original lookups!

Sadly, since we do not have a contiguous set of characters to check (punctuation/CR/LF chars), it adds some complexity. We would only need to check the following chunk combinations on decrypted output:

Upper-case A-Z ASCII {65-90} (We know the first 3 bits are '010', as > 64 and < 91)

0	1	0					

And NOT ASCII (64)

0	1	0	0	0	0	0	0

And the following punctuation/CR/LF bytes {10, 13, 44, 46, 59}

0	0	0	0	1	0	1	0
0	0	0	0	1	1	0	1
0	0	1	0	1	1	0	0
0	0	1	0	1	1	1	0
0	1	1	1	0	0	1	1

Should anything else be read from the decrypted message, we know it is incorrect.

4. Assume cryptosystem numerically assigned alphabet.
   a. If p = 5 and q = 13, list five legal values for d.

Calculate modulus: $n = pq : n = 65$

totient: $\varphi(n) = (p-1)(q-1) = 48$

$Let\ e = \{1 < e < 48\}$ that is coprime/relatively prime with 48

Factors of 48: f = {1, 2, 3, 4, 6, 8, 12, 16, 24, 48}

$e = \{5, 7, 11, 13, 17, 19, 23, 25, 31, 35, 37, 41, 43, 47\}$

$d(e) = e^{-1}\big(mod\ \varphi(n)\big) = \frac{1}{e}\ mod\ 48$

$\boldsymbol{d}(\boldsymbol{e}) =$ **{7, 29, 35, 37, 17, 43, 23, 25, 5, 19, 11, 41, 19, 47}**

   b. If p = 5, q = 31, and d = 37, find e.

Calculate modulus: $n = pq : n = 155$

totient: $\varphi(n) = (p-1)(q-1) = 120$

$Let\ e = \{1 < e < 120\}$ that is coprime/relatively prime with 120, where d = 37

$d = e^{-1}\big(mod\ \varphi(n)\big), e = \frac{1}{d}\ mod\ 120$

$e = \frac{1}{37}\ mod\ 120 = \mathbf{13}$

   c. Using p = 3, q = 11, and d = 9, find e and encrypt ‘‘hello’’.

Calculate modulus: $n = pq : n = 33$

totient: $\varphi(n) = (p-1)(q-1) = 20$

$Let\ e = \{1 < e < 20\}$ that is coprime/relatively prime with 120, where d = 9

$d = e^{-1}\big(mod\ \varphi(n)\big), e = \frac{1}{d}\ mod\ 20$

$e = \frac{1}{9}\ mod\ 20 = \mathbf{9}$

Encrypt “hello”

Message: m = {8, 5, 12, 12, 15}

In general, $x = m^e \, mod \, n$

$$x(m) = \{8^9 \, mod \, 33, 5^9 \, mod \, 33, 12^9 \, mod \, 33, 12^9 \, mod \, 33, \; 15^9 \, mod \, 33\}$$
$$= \{\mathbf{29}, \mathbf{20}, \mathbf{12}, \mathbf{12}, \mathbf{3}\}$$

5. Intelligence Agencies: How they undermine computer security
   Simon Willshire (0491272)

"[R]ecent allegations that the United States government has subverted various cryptographic standards reemphasizes the need to understand avenues for, and defenses against, the deliberate weakening of cryptographic algorithms." [1]

The following sections provide examples of various historic occurrences of encryption standards being deliberately tampered with by various governmental organisations. Most of which are targeting the NSA, which seems to be the forefront of this activity. Proceeding these occurrences describe four various software packages that have yet to be conquered by said governmental organizations (at least publicly stated or leaked to the public). Lastly of which is a mention for software that is suspected to have been breached, but without much evidence either way suggesting its state.

(NSA) Data Encryption Standard (DES)
A well-known encryption standard that was changed without much secrecy that bit length and in turn the overall encryption strength of the standard. The NSA requested the standard be a 48bit key, which originally was set to 64bit, later IBM (the authors of the encryption) and the NSA decided on 56bit key. Obvious at the time, that the NSA was diminishing the brute force difficulty of cracking DES, hardware today is able to crack DES within 21 hours <source>.

(NSA) Lotus Notes 4 (1990s)
Implemented 64 bit keys, 40bit symmetric portion, and 24bit NSA provided RSA encryption. The "feature" was provided in the software's manual, and thus a backdoor sabotage can take place: in this case, the 24bit key can be provided from a "secret key", which can generate the NSA portion of the key. The remaining 40bit sequence can then be brute forced.

(NSA) Dual EC DRBG (FIPS standard for pseudo-random number generator (PRNG)):
Used a dual elliptic curve to deterministically generate random numbers, the curve points are static parameters in the system. Edward Snowden revealed through leaked NSA documents, that the parameters were provided by the NSA who chose them on behalf of NIST, which was originally suggested for the FIPS standard. Through the knowledge of these parameters, a user may be able to predict the future generations of random numbers, any of which can be used to degrade an encryption. It is predicted that the RSA Corporation was paid by the US government to make this algorithm a standard, later used in BSAFE and the Transport Layer Security (TLS) later. This not only lowers encryption standards, as it does not allow un-pocketed companies to rightly make encryption standards, but leaves all encryption wide open to be accessed for those with the prediction knowledge. These methods of random number generation lead us into the next section.

(NSA, GCHQ & CSEC) "Bullrun" HTTPS and SSL Decryption
In 2013, through the leak provided by Edward Snowden - have shown that through the use of government owned supercomputers, typical encryption protocols on the web have been compromised, as well as 4G wireless encryption techniques. Previously, companies such as Microsoft have given pre-encrypted access to their servers' with access to Skype calls, Outlook/Hotmail services [2]. The risk of course is typical to any backdoor access, should a backdoor exists; it only undermines the further security of the system in place, in order for the government to have control of it. With this access, VOIP and common

implementations of banking and shopping systems have been compromised. In addition to support from GCHQ, the Canadian agency CSEC was responsible for a large role in helping the NSA with this program. This was achieved by giving control of an international encryption standard to implement a backdoor decryption technique [4]. Other methods have been employed to decrypt SSL traffic by security organizations, one such method has been named the “poodle flaw”, which was first discovered by Google in 2014. The method makes use of downgrading the use of SSL version to work in 3.0, which can then be compromised. However, web browser software was updated as shortly after the leak was first mentioned, giving a short period or low expected infected users [5].

The Uncrackable Four

TrueCrypt & BitLocker

TrueCrypt was highly recommended as a fully encrypted disk service/software for Windows until the developers pulled the plug on it in May of 2014. The developers now recommend the use of Microsoft’s BitLocker. BitLocker uses 128 bit AES encryption, half the key-size that used for TrueCrypt [6]. As mentioned previously, Microsoft has been asked on numerous occasions to implement backdoors for the NSA, however there has yet to be any news saying that there has been one implemented within BitLocker. It is nearly impossible to decrypt a 256bit AES key, however, as the software is no longer maintained it is a poor example of uncrackable software, but it is uncrackable none the less.

OTR

Off the record protocol was first proposed in 2005, and has established a library which uses short-lived encryption keys which are discarded purely managed in memory. It also makes use of long-term keys to “distribute and authenticate the short lived key” [7]. OTR strives to avoid the use of digital signatures, but relies on user authentication, through the use of their coined “Message Authentication Codes (MACs)”. The encryption process stars off by implementing the Diffie-Hellman algorithm, where each chat member generates a random value, each combined to create a shared secret. The shared secret is then used to authenticate. Next, when messages are sent, each member computes a hash based upon the shared secret, which is encrypted first by AES-128. Once the message has been sent, the receiver must verify that the MAC is correct using the hash of the hash of shared authentication key, which it then uses to descript using the hash of the shared authentication key. The final step of the algorithm is the reassignment of the shared authentication key between members, as well as the hash, and the hash of the hash values used earlier. Once these values have been verified between members, the old values from the last transaction are disposed of. The final step posts the last hash of the hash shared key, this process allows old messages to be forged, but not new messages.

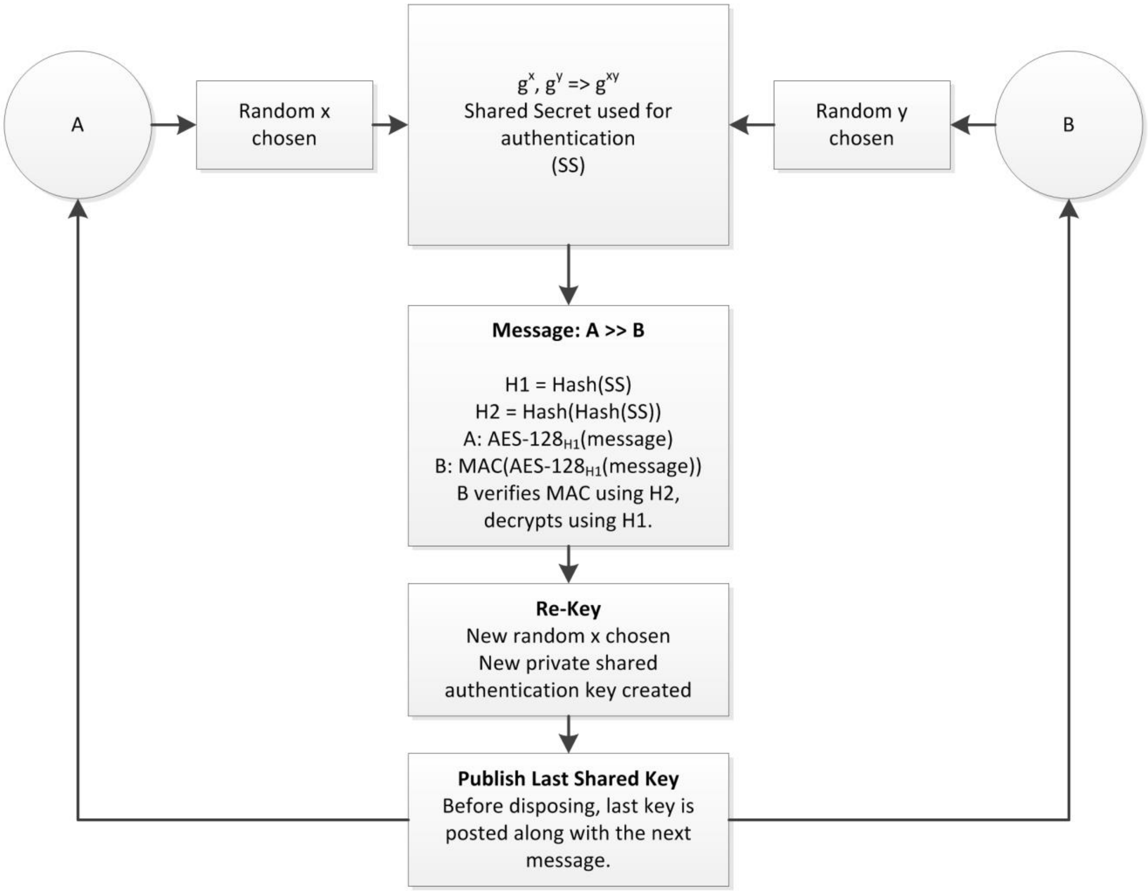
A
Random x chosen
$g^x, g^y \Rightarrow g^{xy}$
Shared Secret used for authentication
(SS)
Random y chosen
B
Message: A >> B
H1 = Hash(SS)
H2 = Hash(Hash(SS))
A: $AES\text{-}128_{H1}$(message)
B: MAC($AES\text{-}128_{H1}$(message))
B verifies MAC using H2,
decrypts using H1.
Re-Key
New random x chosen
New private shared authentication key created
Publish Last Shared Key
Before disposing, last key is posted along with the next message.

Figure 1: A less convoluted explanation shown through diagram of the OTR protocol

OTR overall provides fast and fairly elaborate techniques into casual conversation encryption. It attempts to provide perfect forward secrecy and reputable authentication between chat members - It has yet to be cracked.

TOR

The TOR project is a network which uses virtualised tunnels rather than using direct connections. This tactic allows the network to control the privacy of the interacting nodes. TOR is currently hosted by volunteer servers, further securing the anonymity of the network and its users. The network is extremely difficult to use typical traffic analysis on as with most public networks – ie, the source and destination of a packet is masked and broadcasted through the TOR network to its eventual destination. A typical packet is sent unencrypted to the first TOR node, where it is then propagated to the final destination in an encrypted state, once the packet arrives the destination node/machine is able to then decrypt the contents and provide an appropriate response. The beauty of this system comes from the routing circuits which interchange roughly every 10 minutes, once this time period has been established the network reconfigures itself for new network propagation [8]. TOR is comprised of roughly 76%, a form of the Diffie-Hellman key (1024bit length) encryption. However newer versions of TOR use an elliptical key which are much more difficult to crack. As of now, there has been no news on the cracking (Government or not) of the TOR network.

Zoho

Zoho is a company that provides encryption services to all manner of security users. At an encryption level, it provides 128 and 256bit AES sent through SSL encryption for emails. This method of transfer is incredibly hard to crack, and near impossible using brute force techniques. Evidently, the longer the key encrypted (with linear increase in complexity), creates and exponential increase in decryption/cracking difficulty [9]. There have been no reports of Zoho being cracked, nor contacted in hopes of allowing a backdoor into their software.

Aside: Pretty Good Privacy (PGP)

Without going into much detail, PGP freeware version is an open source encryption standard, which has caused some news of late in regard to being compromised by the NSA. The legality problems originally arose from the RSA foundation being purchased by the NSA. Allegedly PGP was rumoured to have had a backdoor put in place as of being purchased, however as open source software, the general public is able to examine its contents and deem it secure. If PGP was compiled by a non-authentic source revision, the binaries are "identified as forbidden by the license". PGP in general used to be known for being uncrackable, however it looks as though the project has been abandoned by many users due to distrust in the fact of being sold to the NSA through the acquisition of the RSA foundation. Overall, this encryption software should still be considered as uncrackable, however the politics behind it is a little fishy.

Sources

[1] https://www.schneier.com/paper-weakening.html
[2] http://www.theguardian.com/world/2013/jul/11/microsoft-nsa-collaboration-user-data
[3] http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security
[4] http://www.huffingtonpost.ca/2013/09/11/csec-nsa-encryption_n_3907748.html
[5] http://www.cnet.com/news/google-exposes-poodle-flaw-in-web-encryption/
[6] http://www.tomshardware.com/reviews/bitlocker-truecrypt-encryption,2587.html
[7] https://otr.cypherpunks.ca/otr-codecon.pdf
[8] https://www.torproject.org/about/overview
[9] https://www.zoho.com/security.html
[10] http://www.cam.ac.uk.pgp.net/pgpnet/

6. Calculate the cost of watching a 2 hour video in Barbados, Mali and Laos, assuming 312x390 res, with no cropping/aspect ratio adjustments.

Estimate the bandwidth requirements for you a watch a video with audio (assumed stereo output):

Stereo audio: 384 kbps
1080p video: 8 Mbps
Total 1080p bandwidth to scale = 8,384 kbps

Scaled Video: (1920 x 1080) to (312 x 390):

$$= \frac{8{,}384 kbps}{\left(\frac{1920}{312}\right)\left(\frac{1080}{390}\right)} = 492\ kbps$$

Assuming exactly 2 hour video streaming: $b = \frac{492(60)(60)(2)}{8} = 442{,}800\ kB = 432.421875\ MB$
According to bell roaming charges:

Barbados: $8/MB	**Mali: $16/MB**	**Laos: $12/MB**
Cost: $(8)432.42 = \$\mathbf{3{,}459.38}$	Cost: $(16)432.42 = \$\mathbf{6{,}918.75}$	Cost: $(12)432.42 = \$\mathbf{5{,}189.06}$

Grand total of absorbent roaming charges: **$15,567.19**