Tiny Invaders: New Threats from Cables We Take for Granted

BSIDES PORTLAND

We Will Talk About

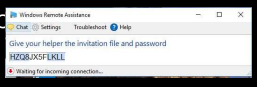
1. The Good – A System On A Chip embedded In A Cable
2. (Protecting against) The Bad Chip embedded in a Cable
3. The Ugly – What an attacker can do
4. Encrypt Everything
5. Rolling Your Own Encryption
6. Captain Janeway's Encryption
7. Closing

2

We are ALL Tech Support …

Tech Support Phone call …

Click the "Get Help" Ico



Click "Send"

Click "Yes" to allow me …

Click the box and "Yes"

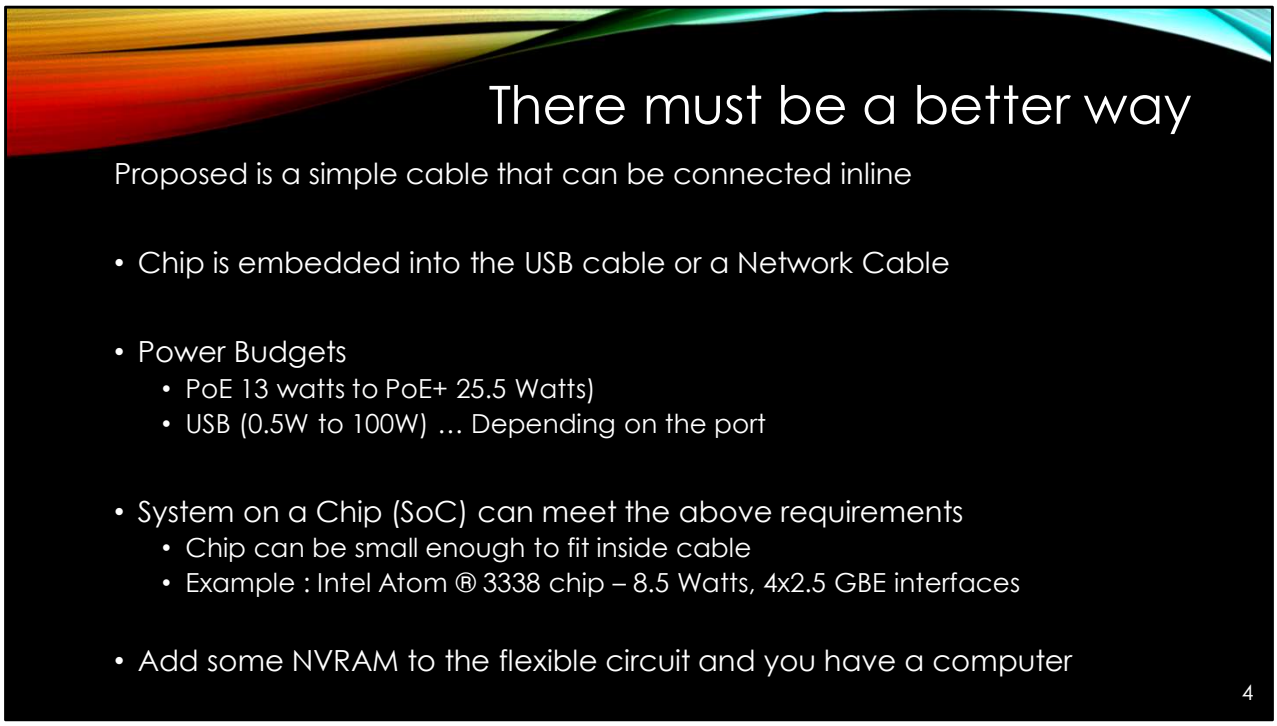
NOOO … NOOO !!!!!!!

3

My inspiration for this cable … Phone Call with parents to connect to their computer

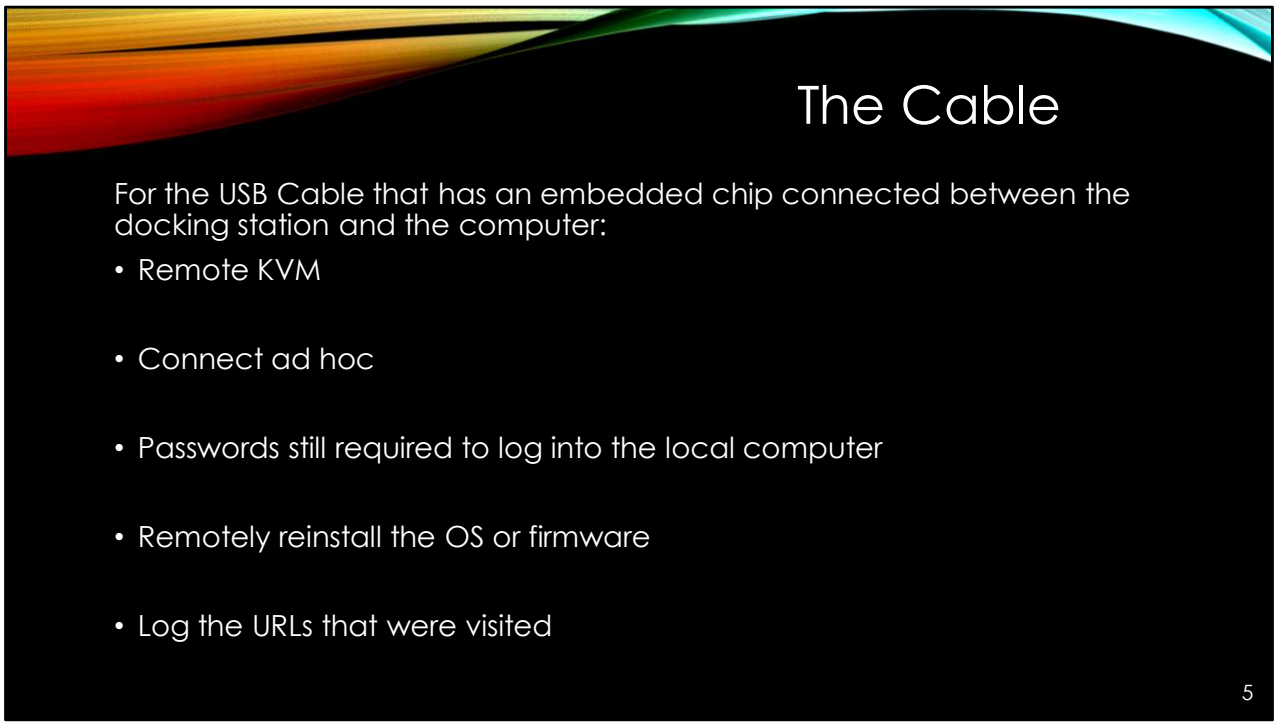
1) Microsoft Windows 10 get help from friend
2) VNC
3) Chrome Remote
4) Go To Meeting
5) Etc …

… But installing software on a computer to allow remote access can allow attackers to remotely access the computer also if the intermediary site is somehow compromised!!! ???

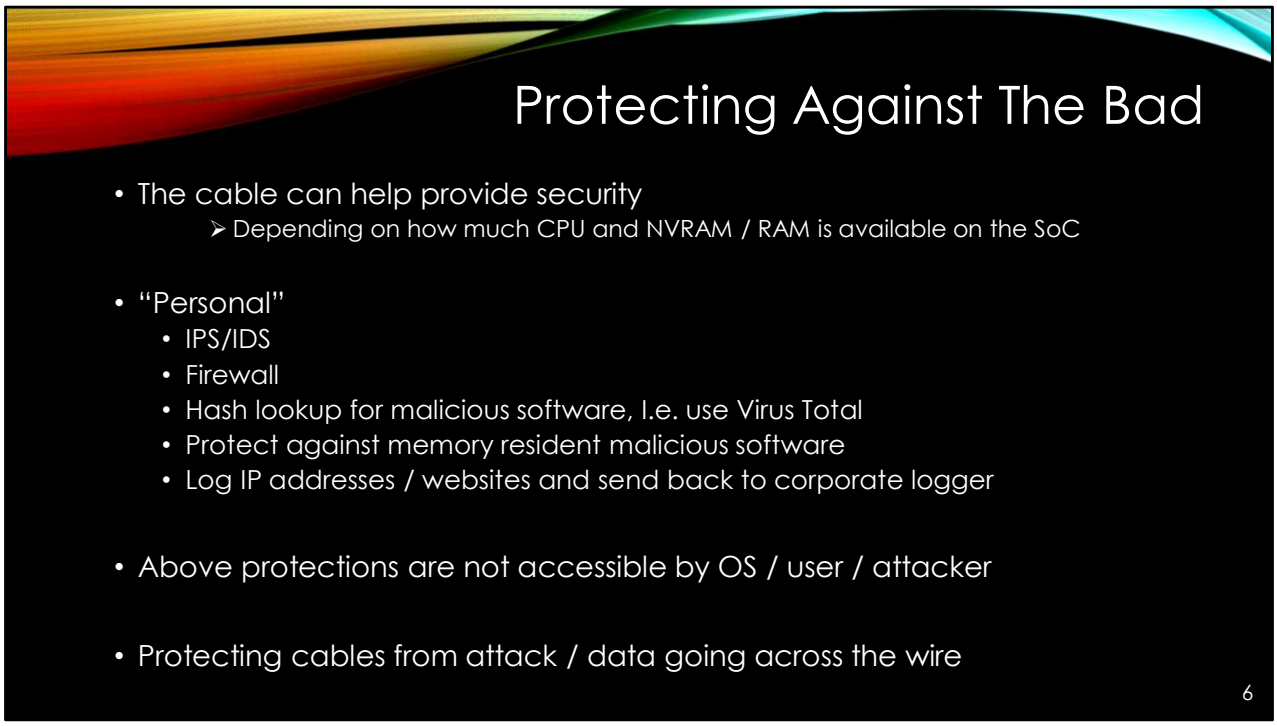


1) Overview
Chip is embedded into the cable – Ethernet or USB
Power Budgets – PoE (13 watts to 25.5 Watts), USB (0.5W to 100W)
Heat Dissipation via the shielded wire, chips mounted on flexible circuits wrapped around the wire or inside the connectors
Apple Lightning → USB https://en.wikipedia.org/wiki/Lightning_(connector)
USB-C already has an embedded chip – What is on that chip?
https://en.wikipedia.org/wiki/USB-C#Cables "They are electronically marked cables that contain a chip with an ID function based on the configuration channel and vendor-defined messages (VDM) from the USB Power Delivery 2.0 specification."
Low power USB devices – 0.5 W up to high power – 100W -
https://en.wikipedia.org/wiki/USB#Low-power_and_high-power_devices
As with all specs it is … complicated -
https://en.wikipedia.org/wiki/USB_(Physical)#Power_Delivery_(PD) – Speaks about power allowance / requests
Different SoC's and their power requirements -
https://en.wikipedia.org/wiki/System_on_a_chip
Die size - https://en.wikipedia.org/wiki/List_of_Intel_Atom_microprocessors -
Diamondville 45 nm - Die size: 25.96 mm² (3.27mm×7.94mm, 0.128"x0.314")

Intel is now at a 10nm size as of 2017, you can do the math on how the above die size has shrunk - http://fpga.org/2017/03/29/intel-10nm-the-wind-in-our-sails/
Intel specs from Intel Website / PDF - https://www.intel.com/content/www/us/en/processors/atom/atom-c3000-family-brief.html



2) The Good USB (cable) that connects the computer to the docking station:
Can be used as a remote KVM – The tech support we ALL do – Mom and Dad
Allows the remote access to get into BIOS / Complex operations (if the docking stations gets DHCP and it powers the USB). Essentially an OOB / Poor mans iLO
Always on, no software to install assuming the computer is turned on
Passwords are needed to get into the locked system, same as at the keyboard. Brute forcing would be painful assuming a good password is chosen
Log URLs - "What did you install?" – "I didn't install anything" (check the logs)



3) (Protecting against) The Bad

→ How much of the below depends on how much CPU and NVRAM is available to the cable

IPS / IDS

Firewall

Offsite Worker protection (I.e. send new firewall or IPS rules to chip / cable that you sent home with the user outside of the corporate environment), cheapter than a full scale IDP/IPS per employee

Rules / monitoring cannot be changed by malicious software on the host machine, controlled by you, the administrator

Check incoming executables before passing to the local computer, hold until compare with VirusTotal is complete and pass along if not malicious

Not controlled by the user / onboard software so cannot be deleted from the system

Logging of IP's / websites visited

Military grade hardening of cables:

Pressurized Ethernet Cable Runs

Point to Point encryption devices

3) (Continued)
Monitoring the data ingressing and exiting your network is more important than ever, are you positive that someone hasn't plugged a rogue Wireless or wired ethernet tap into your network already?
Do you passively monitor your users DNS requests for "strange" websites?
E.g. https://azure.microsoft.com/en-us/blog/heuristic-dns-detections-in-azure-security-center/
https://www.helpnetsecurity.com/2013/05/28/dns-anomaly-detection-defend-against-sophisticated-malware/
Be aware of supply chain attacks:
Cables left in public places or mailed to you
Do you know if your supplier is giving you 1 in 50 cables with a chip? If you have a large project with 10,000 cables 1:50 cables might get one installed in a PoE switch
What about Transceivers that you put into your networking equipment? They could also be compromised
Do you have a meeting room or a lobby outside of your locked doors at your company that are not monitored?
Somewhere an attacker could drop off a compromised cable? Attach an ethernet tap / Raspberry Pi into your network surreptitiously?

As an aside …

- Bloomberg speaking about supply chain attacks
  - (I am skeptical)

- MAYBE in 5 years
- BUT … What other devices can contain these chips?

"Paranoia is just reality on a finer scale" - 1995 film Strange Days

8

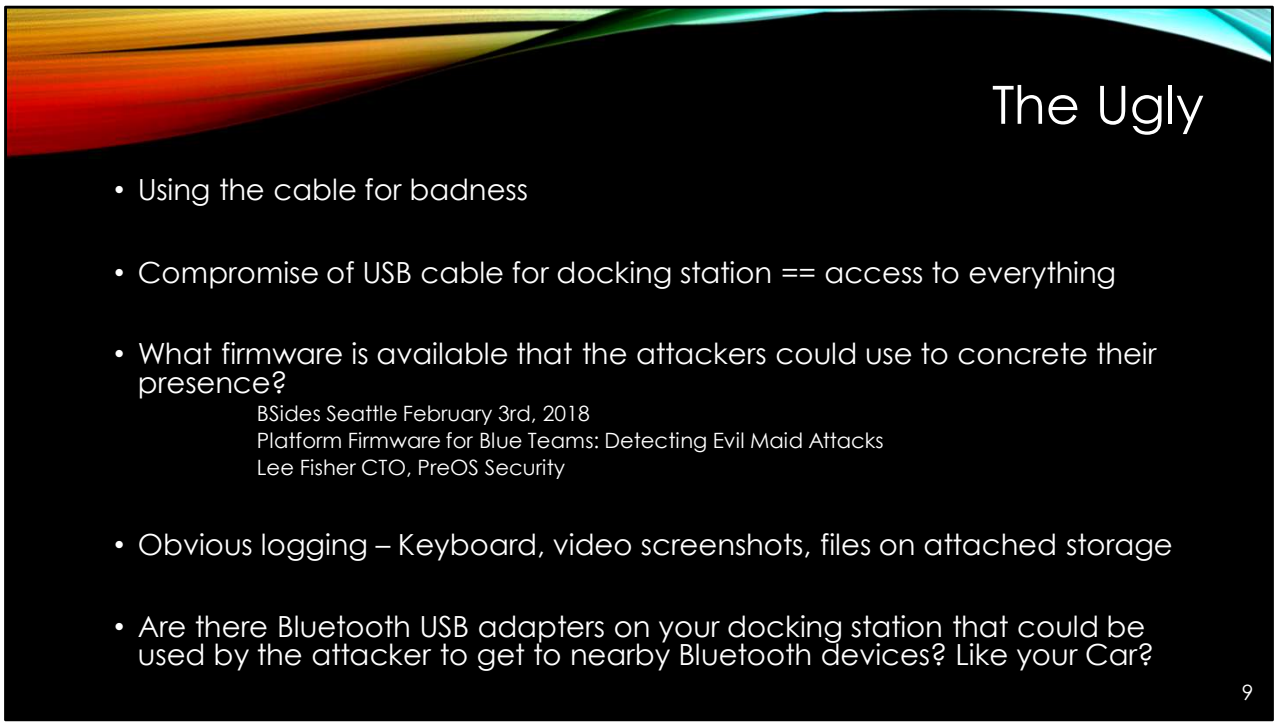
The original Bloomberg Story:
https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies

An implanted chip in a server is possible, but the size of a pencil tip? I am skeptical that the technology is that small … But …

An example of how this attack might work:
https://www.electronicdesign.com/embedded-revolution/how-hack-server-motherboard

USB Key Logger hidden in a cable "Bad USB":
http://mg.lol/blog/badusb-cables/

Chips can be hidden in Docking Stations, USB Hubs, anything connected to your computer



4) The Ugly

Obviously, a tool that is used for good can also be subverted to be used for the bad

Ethernet – The attacker can track anything on the wire FOR THAT ONE USER, clear text passwords anybody?

USB – If the USB is between the laptop and docking station the attacker owns everything that is on that USB bus → Keyboard, Video, Attached storage, Attached DVD drives, Network adapters, Blue tooth keyboard, Etc …

Exploit of video from HDMI connected port – There is a small computer in your monitor:

https://twitter.com/xipitersec/status/764188263858876416

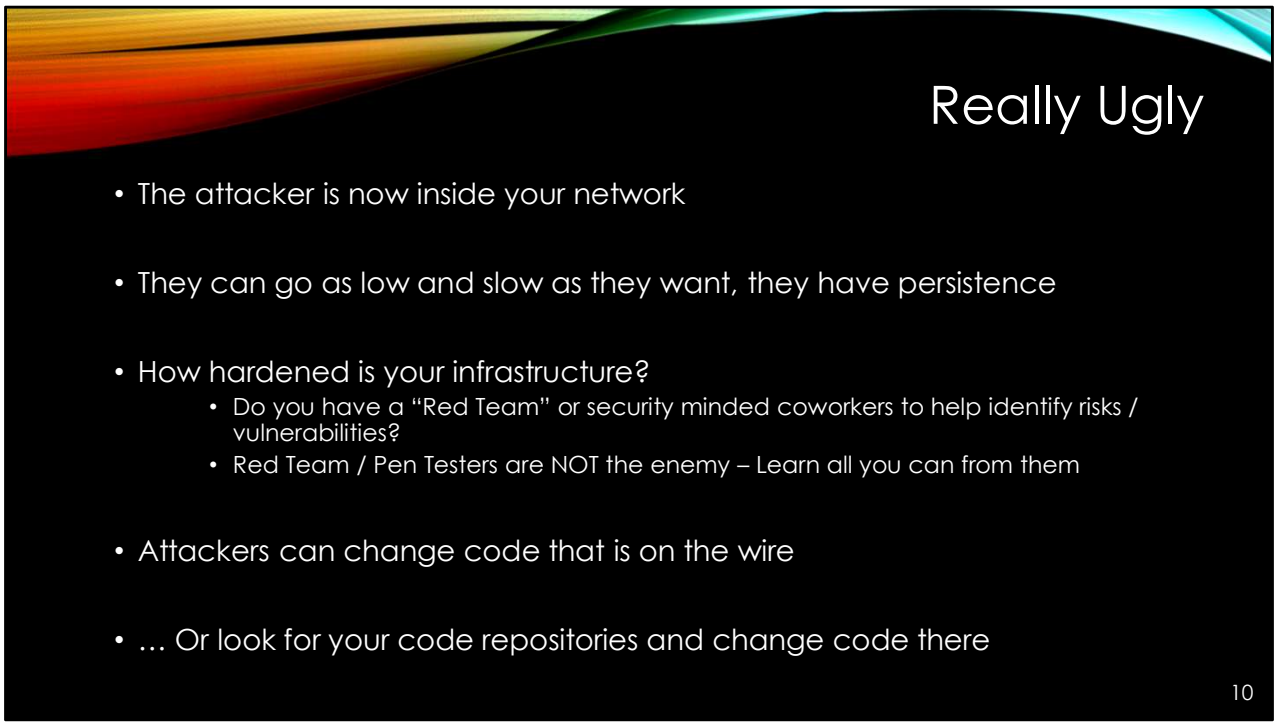
Access to firmware of any connected USB devices

https://firmwaresecurity.com/tag/bsidesseattle/

https://firmwaresecurity.files.wordpress.com/2018/02/bsidesseattle2018-fisher-defending-firmware.pdf

KVM control

Keyboard Logging

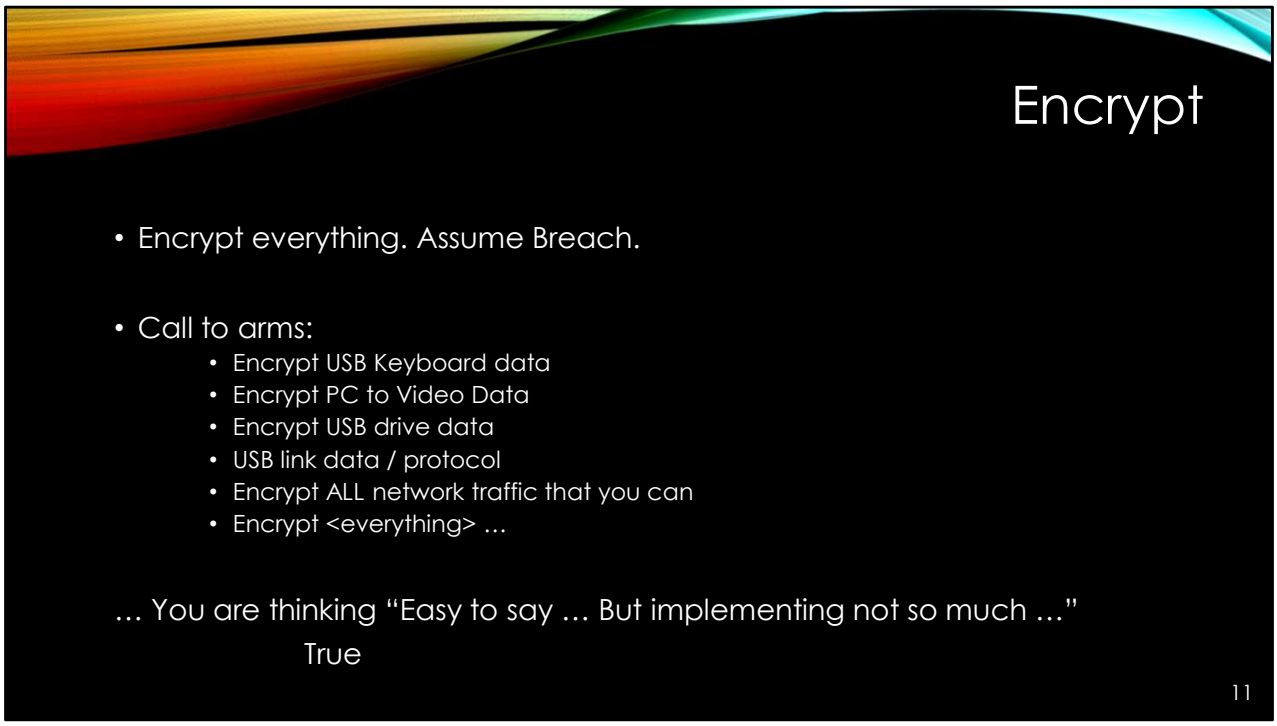


4) … (Continued)
Standard stages of attacks with an "extra" one:
Phase 1: Reconnaissance
Phase 2: Initial compromise
Phase 3: Command & control
Phase 3.1: After the cable is installed MORE Reconnaissance to see what internal resources can be accessed
Phase 4: Lateral movement
Phase 5: Target attainment
Phase 6: Exfiltration, corruption, and disruption
The attacker now has access internally to your network / devices on your network
Are you "Hard and Crunchy" on the outside but "soft and chewy" on the inside?
Attacker can go low and slow, look for the user to access open SMB shares and look for secrets on those network shares
You don't have ANY open shares on your network, correct?
Your users don't open personal / computer shares to each other, correct?
Do you have / have you hired a Red Team?:
https://www.tripwire.com/state-of-security/risk-based-security-for-executives/connecting-security-to-the-business/red-team-v-blue-team-they-are-in-fact-

one-the-purple-team/
Change code (either ingress or egress) as it passes over the wire, intercept, hold the file till complete and make changes to insert attack software:

- Incoming .exe files
- Incoming .zip or .rar files that have executables inside, change those
- Incoming .sh or .ps1 files or any kind of scripts
- Now the attacker is on the machine through one of the pieces of software that was downloaded

Look for code repositories and add “extra” code into your repository



What part of "Encrypt Everything" is not clear?:
https://www.beencrypted.com/ultimate-encryption-guide/

USB Link Protocol - http://www.techdesignforums.com/practice/technique/usb-3-0-link-layer/

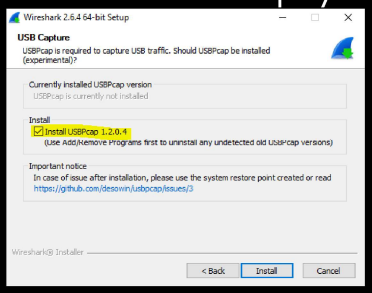
Design and Verification of USB 3.0 Link Layer (LTSSM) - https://pdfs.semanticscholar.org/a4bf/64df084659820daedaf02f47fb5ae2bf411f.pdf

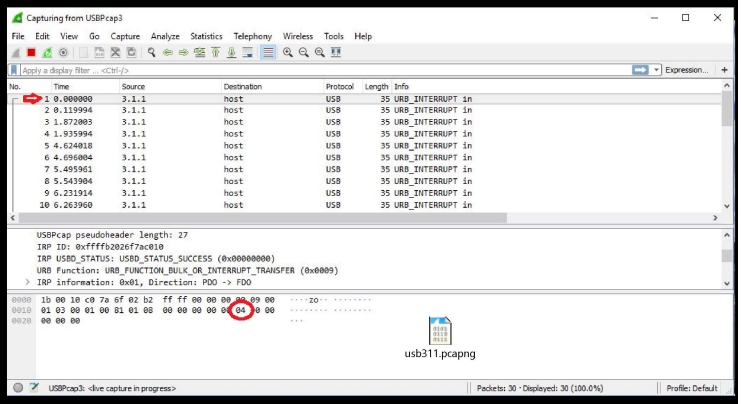
The attacker can have access to ANYTHING connected to your USB hub. You must assume that there is a man-in-the-middle already

This is NOT an easy task. Everybody who develops software and firmware will have to work on this problem "In their free time" …

Dump the traffic on your USB

You can dump your USB traffic easily using Wireshark





And you can see per the dump and the USB HID usage table that I typed an "a"

12

If you are interested in seeing what data goes across the USB bus you can dump using Wireshark

USB HID table:
http://www.freebsddiary.org/APC/usb_hid_usages.php

Rolling your own encryption

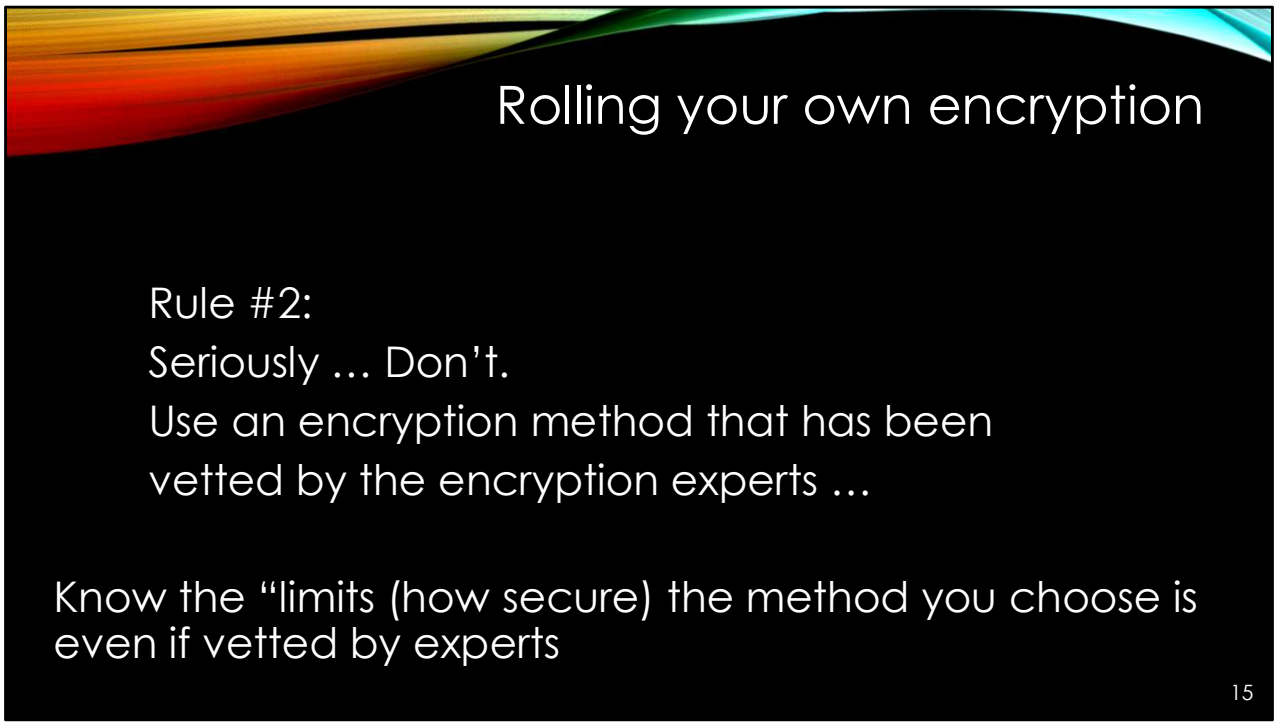
Steps for creating your own encryption protocol

13

Rolling your own encryption

Rule #1:
Don't

14

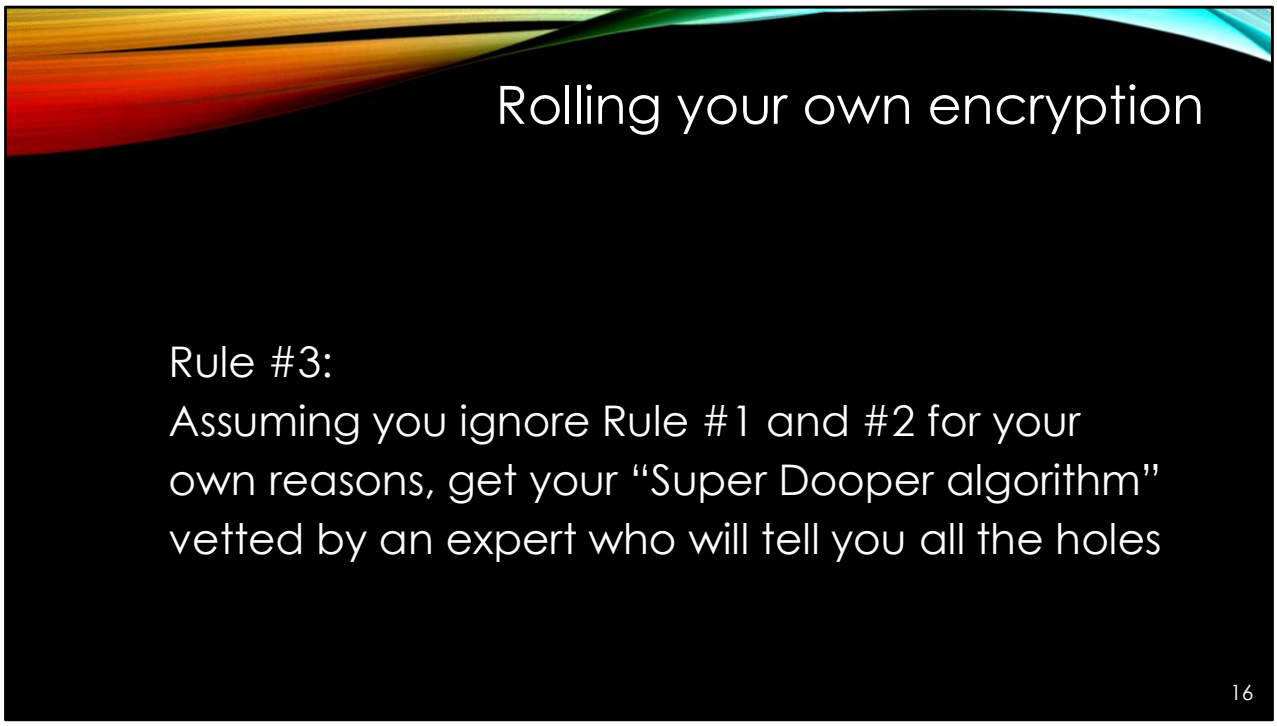


Limits of how long an encryption key is good for. Example the DES standard:
In DES 56-bit key 1970 considered "Secure".

Per https://en.wikipedia.org/wiki/Data_Encryption_Standard :
"the Electronic Frontier Foundation's DES cracker in 1998 that demonstrated that DES could be attacked very practically, and highlighted the need for a replacement algorithm"

MD5:
https://en.wikipedia.org/wiki/MD5
'MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function MD4,[4] and was specified in 1992 as RFC 1321.'
'As of 2010, the CMU Software Engineering Institute considers MD5 "cryptographically broken and unsuitable for further use",'

Know the limits of your technology, allow for that technology to be upgraded easily in your code, put the encryption code in its own routine.



The below method, which is more obfuscation than encryption, was looked at by an expert. They very kindly told me that the method was less than comprehensive and not difficult to break.

… And I rolled my own

- For a different project implemented a simple peer-to-peer text chat, encrypted
- Devised own encryption as opposed to implementing something like ECC:

https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

Theory [edit]

For current cryptographic purposes, an *elliptic curve* is a plane curve over a finite field (rather than the real numbers) which consists of the points satisfying the equation

$$y^2 = x^3 + ax + b,$$

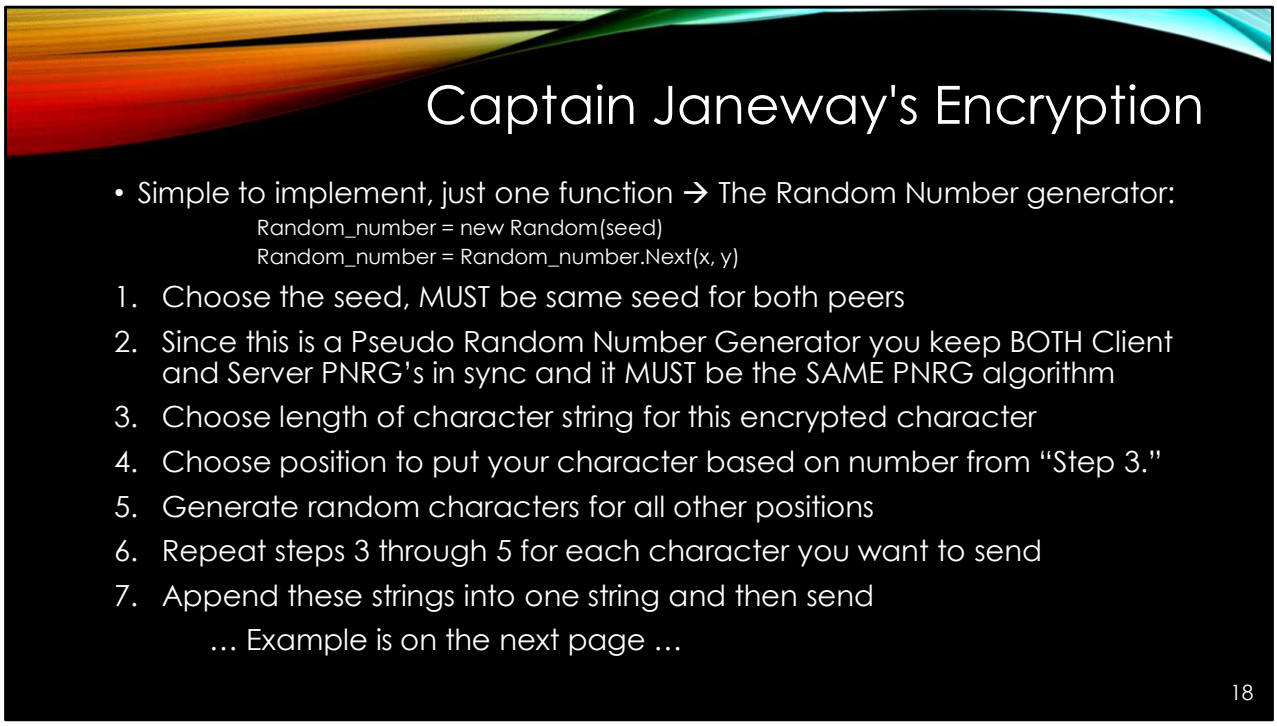
along with a distinguished point at infinity, denoted ∞. (The coordinates here are to be chosen from a fixed finite field of characteristic not equal to 2 or 3, or the curve equation will be somewhat more complicated.)

This set together with the group operation of elliptic curves is an Abelian group, with the point at infinity as identity element. The structure of the group is inherited from the divisor group of the underlying algebraic variety.

$$\mathrm{Div}^0(E) \to \mathrm{Pic}^0(E) \simeq E,$$

17

Yeah … Right. I just wanted a simple algorithm



Captain Janeway's Encryption description

PNRG's are exactly as described, they *LOOK* like random number generators but when you put start with the same seed, they give out the same number for the .Next function

A crypto professional looked and very gently gave me excellent feedback. The issue is with this obfuscation is that if the attacker has your code (and they will if it is in the firmware), the algorithm just like the "Enigma" machine. The obfuscation can be broken by trying all the combinations for the seed:
https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma

This is why I don't do electrical installations except the simplest ones at my own house ☺ …

Capt. J's Encryption example

- String to encrypt and send is "Hello World" (original, I know), max random length of 50:

Charter position = "i", Rand1 = "Number of Random Characters", Rand2 = "Position of 'encrypted' charter in Rand 1 string", String = string generated

String to send = "Hello World" (length = 11 characters, Characters 0 through 10)

i=0, Rand1 32, Rand2 20, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;

i=1, Rand1 12, Rand2 5, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72

i=2, Rand1 49, Rand2 40, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90

i=3, Rand1 28, Rand2 13, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/)

i=4, Rand1 36, Rand2 32, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh
X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&

i=5, Rand1 22, Rand2 9, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh
X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&]j7I6|8_'!/S`h_to;Pik

i=6, Rand1 24, Rand2 11, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh
X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&]j7I6|8_'!/S`h_to;PikMBV"8CD"}tWHK`yIAL6BQO>v

i=7, Rand1 22, Rand2 13, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh
X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&]j7I6|8_
'!/S`h_to;PikMBV"8CD"}tWHK`yIAL6BQO>vV0[WS]W0Ag']oy ;a}0]7<

i=8, Rand1 49, Rand2 47, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh
X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&]j7I6|8_
'!/S`h_to;PikMBV"8CD"}tWHK`yIAL6BQO>vV0[WS]W0Ag']oy ;a}0]7<Ex#SHyw.J,H`3fHI6o;?2AaP5&mP.81:V65]O*.4GX[QQ=r(x

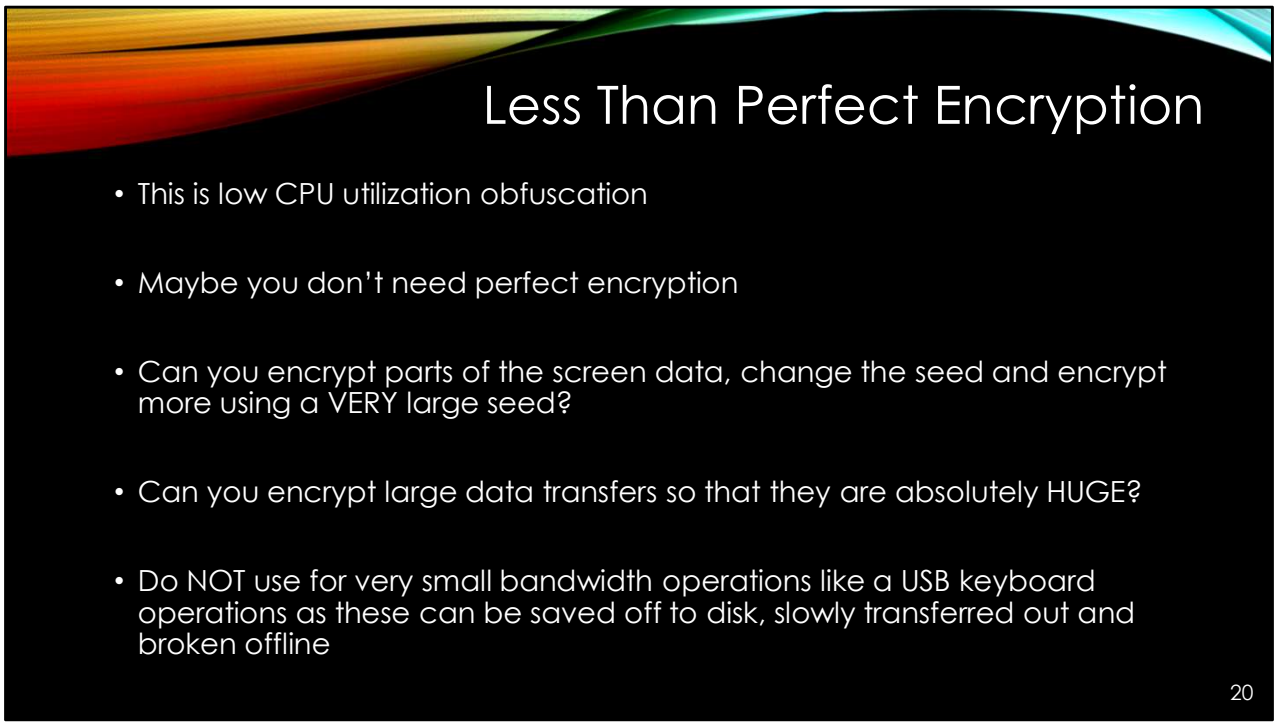
i=9, Rand1 28, Rand2 21, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh
X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&]j7I6|8_
'!/S`h_to;PikMBV"8CD"}tWHK`yIAL6BQO>vV0[WS]W0Ag']oy ;a}0]7<Ex#SHyw.J,H`3fHI6o;?2AaP5&mP.81:V65]O*.4GX[QQ=r(xFF!]f#6P=B+G}1x.)zQhlw,`'uWd

i=10, Rand1 9, Rand2 9, String pnXI).K7 o|koL7XP{2HKcGRYe>-qw;;>Wnke{yT%"72a'1n76D/uH;|bh X=uSeEPrGjS!\F`/@Zw[%_)#1$p6G.Rl90b?\7G03h#b5elf49xBh,xw4rOs/){9n(z{mbCPKpu%>m*M2sWJN#YQNK,A7o'Ls&]j7I6|8_'!/S`h_to;PikMBV"8CD"}tWHK`yIAL6BQO>vV0[WS]W0Ag']oy ;a}0]7<Ex#SHyw.J,H`3fHI6o;?2AaP5&mP.81:V65]O*.4GX[QQ=r(xFF!]f#6P=B+G}1x.)zQhlw,`'uWdu{sxHpi9d

19

This works for ASCII characters or for pure data. The "filler" characters need to be the same as the characters you are trying to obfuscate. If your obfuscated characters are all ASCII then only use filler characters that are ASCII, if all your characters is data that is 0 through 255 then use random values of 0 through 255 for your filler.

This is an ASCII example so that it is easier to show as an example.



The case for Less-Than-Perfect Encryption

Low CPU utilization for this method, is it secure "enough" for your purposes?

This encryption could be used in conjunction with “real” encryption to change the PRNG seed on a regular basis

I.e. spike the CPU using “real” encryption for a new PRNG seed but not constantly bogging down the CPU

More obfuscation than true encryption

Stream encryption – You can encode a handful of characters, send then, do more, send them, etc.

Block encryption– Encode a block of characters that fits in “X” characters, write to disk, possibly add in filler characters as needed to obfuscate further

The "seed" should be strong encryption via asymmetric algorithm that is somehow negotiated between the two machines performing the exchange

Do NOT Roll your own PRNG (Pseudo Random Number Generator). See previous comments about rolling your own encryption.

Needs the same EXACT PRNG on both ends of the connection

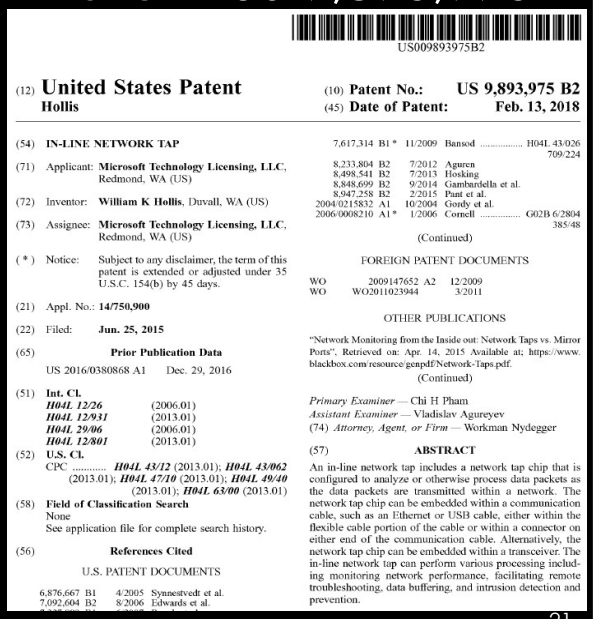
Your code in the firmware is not secret, so the algorithm can be reverse engineered

Can someone create a software Enigma machine to figure out what you are obfuscating?

Can they run software on the host computer to offload the encryption because of large amounts of bandwidth and get the goods locally for later exfiltration?
Can Quantum bit easily decode this algorithm? Is it open to Markov chain analysis?
What kind of data needs "strong" encryption rather than "just enough"?

Patent US 9,893,975

- If you thought that creating these cables might be a new business for you …
  - Note: I Am Not A Lawyer (IANAL), you will have to figure out how to work with Microsoft. Don't ask me, I can't help.
  - (… And I am SURE that Nation states wouldn't infringe on this patent … Right? And countries that don't have intellectual property agreements with the US would never build this, right?)
  - Hence why we need to encrypt everything



US009893975B2

(12) **United States Patent**
Hollis

(10) **Patent No.: US 9,893,975 B2**
(45) **Date of Patent: Feb. 13, 2018**

(54) **IN-LINE NETWORK TAP**

(71) Applicant: **Microsoft Technology Licensing, LLC**, Redmond, WA (US)

(72) Inventor: **William K Hollis**, Duvall, WA (US)

(73) Assignee: **Microsoft Technology Licensing, LLC**, Redmond, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 45 days.

(21) Appl. No.: **14/750,900**

(22) Filed: **Jun. 25, 2015**

(65) **Prior Publication Data**

US 2016/0380868 A1 Dec. 29, 2016

(51) **Int. Cl.**
H04L 12/26 (2006.01)
H04L 12/931 (2013.01)
H04L 29/06 (2006.01)
H04L 12/801 (2013.01)

(52) **U.S. Cl.**
CPC ***H04L 43/12*** (2013.01); ***H04L 43/062*** (2013.01); ***H04L 47/10*** (2013.01); ***H04L 49/40*** (2013.01); ***H04L 63/00*** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,876,667 B1 4/2005 Synnestvedt et al.
7,092,604 B2 8/2006 Edwards et al.
7,617,314 B1 * 11/2009 Bansod H04L 43/026 709/224
8,233,804 B2 7/2012 Aguren
8,498,541 B2 7/2013 Hosking
8,848,699 B2 9/2014 Gambardella et al.
8,947,258 B2 2/2015 Pant et al.
2004/0215832 A1 10/2004 Gordy et al.
2006/0008210 A1 * 1/2006 Cornell G02B 6/2804 385/48
(Continued)

FOREIGN PATENT DOCUMENTS

WO 2009147652 A2 12/2009
WO WO2011023944 3/2011

OTHER PUBLICATIONS

"Network Monitoring from the Inside out: Network Taps vs. Mirror Ports", Retrieved on: Apr. 14, 2015 Available at; https://www.blackbox.com/resource/genpdf/Network-Taps.pdf.
(Continued)

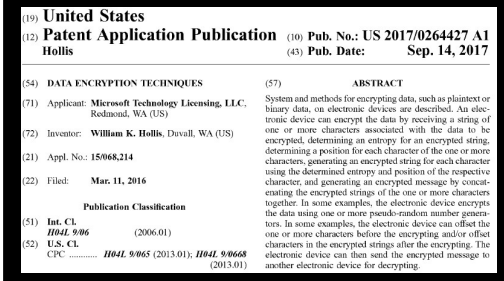
Primary Examiner — Chi H Pham
Assistant Examiner — Vladislav Agureyev
(74) *Attorney, Agent, or Firm* — Workman Nydegger

(57) **ABSTRACT**

An in-line network tap includes a network tap chip that is configured to analyze or otherwise process data packets as the data packets are transmitted within a network. The network tap chip can be embedded within a communication cable, such as an Ethernet or USB cable, either within the flexible cable portion of the cable or within a connector on either end of the communication cable. Alternatively, the network tap chip can be embedded within a transceiver. The in-line network tap can perform various processing including monitoring network performance, facilitating remote troubleshooting, data buffering, and intrusion detection and prevention.

The Good Part …

- But Wait … There's More …
- If you liked Captain Janeway's encryption have at it:
  ABANDONED (Again, IANAL so you should probably talk to one first)
  `http://www.digital.net/~gandalf/CaptJ_Enc.txt`



(19) **United States**
(12) **Patent Application Publication** (10) **Pub. No.: US 2017/0264427 A1**
Hollis (43) **Pub. Date: Sep. 14, 2017**

(54) **DATA ENCRYPTION TECHNIQUES**

(71) Applicant: **Microsoft Technology Licensing, LLC**, Redmond, WA (US)

(72) Inventor: **William K. Hollis**, Duvall, WA (US)

(21) Appl. No.: **15/068,214**

(22) Filed: **Mar. 11, 2016**

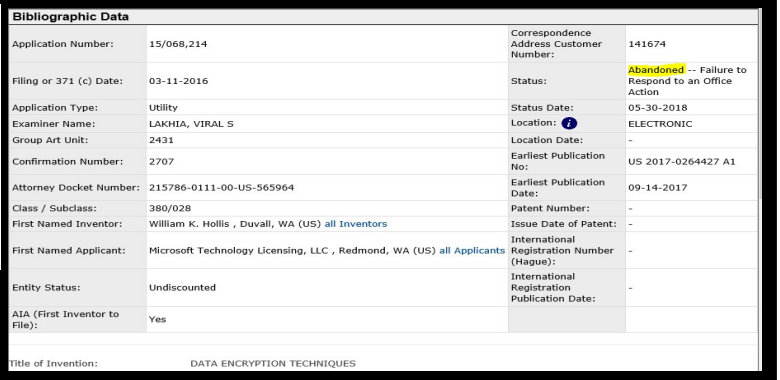
Publication Classification

(51) **Int. Cl.**
H04L 9/06 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 9/065*** (2013.01); ***H04L 9/0668*** (2013.01)

(57) **ABSTRACT**

System and methods for encrypting data, such as plaintext or binary data, on electronic devices are described. An electronic device can encrypt the data by receiving a string of one or more characters associated with the data to be encrypted, determining an entropy for an encrypted string, determining a position for each character of the one or more characters, generating an encrypted string for each character using the determined entropy and position of the respective character, and generating an encrypted message by concatenating the encrypted strings of the one or more characters together. In some examples, the electronic device encrypts the data using one or more pseudo-random number generators. In some examples, the electronic device can offset the one or more characters before the encrypting and/or offset characters in the encrypted strings after the encrypting. The electronic device can then send the encrypted message to another electronic device for decrypting.



Bibliographic Data

Application Number:	15/068,214	Correspondence Address Customer Number:	141674
Filing or 371 (c) Date:	03-11-2016	Status:	Abandoned -- Failure to Respond to an Office Action
Application Type:	Utility	Status Date:	05-30-2018
Examiner Name:	LAKHIA, VIRAL S	Location:	ELECTRONIC
Group Art Unit:	2431	Location Date:	-
Confirmation Number:	2707	Earliest Publication No:	US 2017-0264427 A1
Attorney Docket Number:	215786-0111-00-US-565964	Earliest Publication Date:	09-14-2017
Class / Subclass:	380/028	Patent Number:	-
First Named Inventor:	William K. Hollis , Duvall, WA (US) all Inventors	Issue Date of Patent:	-
First Named Applicant:	Microsoft Technology Licensing, LLC , Redmond, WA (US) all Applicants	International Registration Number (Hague):	-
Entity Status:	Undiscounted	International Registration Publication Date:	-
AIA (First Inventor to File):	Yes		

Title of Invention: DATA ENCRYPTION TECHNIQUES

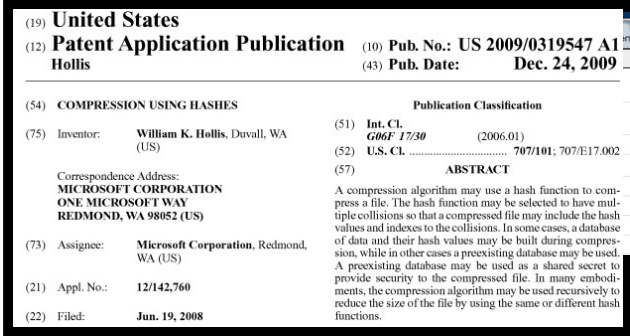
22

See the Patent Office for the above information on Application Number: 15/068,214 (Otherwise known as US 2017-0264427 A1):
http://pdfaiw.uspto.gov/.aiw?docid=20170264427

Client and server all in one program:
http://www.digital.net/~gandalf/CaptJ_Enc.txt

As an aside …

- If you happen to have a 1028 bit quantum bit computer waiting for a project …
- "COMPRESSION USING HASHES", a proposal for recursive compression:
  `http://pdfaiw.uspto.gov/.aiw?docid=20090319547`



(19) **United States**
(12) **Patent Application Publication**
Hollis

(10) **Pub. No.: US 2009/0319547 A1**
(43) **Pub. Date: Dec. 24, 2009**

(54) **COMPRESSION USING HASHES**

(75) Inventor: **William K. Hollis**, Duvall, WA (US)

Correspondence Address:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052 (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA (US)

(21) Appl. No.: **12/142,760**

(22) Filed: **Jun. 19, 2008**

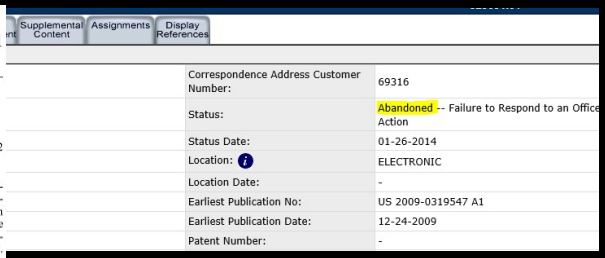
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/101**; 707/E17.002

(57) **ABSTRACT**

A compression algorithm may use a hash function to compress a file. The hash function may be selected to have multiple collisions so that a compressed file may include the hash values and indexes to the collisions. In some cases, a database of data and their hash values may be built during compression, while in other cases a preexisting database may be used. A preexisting database may be used as a shared secret to provide security to the compressed file. In many embodiments, the compression algorithm may be used recursively to reduce the size of the file by using the same or different hash functions.



Supplemental Content | Assignments | Display References

Correspondence Address Customer Number:	69316
Status:	Abandoned -- Failure to Respond to an Office Action
Status Date:	01-26-2014
Location:	ELECTRONIC
Location Date:	-
Earliest Publication No:	US 2009-0319547 A1
Earliest Publication Date:	12-24-2009
Patent Number:	-

I don't have a Quantum bit computer, so I cannot test out the implementation, but I would appreciate it if someone would:
http://pdfaiw.uspto.gov/.aiw?docid=20090319547

Likewise abandoned … Application Number: 12/142,760

For a copy of these slides (and all my notes and links)
See my OneDrive:

https://1drv.ms/f/s!AsoLDJx_szsuc_zSFFU0B1gKkFc

Folder:

BSides_Portland/BSidesPDX_Tiny_Invaders_Final.pptx

BSides_Portland/BSidesPDX_Tiny_Invaders_Final.pdf