technology for the creative

Phil Crawley
@IsItBroke on Twitter

Tech Breakfast – Encryption, a primer

23rd March, Soho Screening Rooms, London.

Starting with the fundamentals of cryptography (difference between symmetric & public-key etc), we will look at the encryption techniques used in the Enigma machine (with a genuine, working example) through to HDCP and the other forms of content protection used in our industry:

- Symmetric vs Public-key crypto
- Examples – DES, AES/Rijndael etc
- HDCP, Blueray and the MPAA etc
- PCoIP – security of KVM extenders

Cryptography, an introduction

The Internet as well as many file-systems depend on cryptography to keep information secure;

- Shopping or banking websites – need for confirming identity and securing traffic
- Content Protection mechanisms for baseband video; HDCP and DCinema
- Securing files on a hard drive to prevent data loss/theft.
- Secured remote desktops in KVM-over-IP extenders.

There are essentially three technologies used to achieve this;

- Symmetric Cryptography – the same key encrypts as well as decrypts the data
- Asymmetric (AKA “Public Key”) Crypto – uses separate encrypting and decrypting keys
- Hashing – mathematical functions that derive a unique number for a file

Encrypting text

"We attack at dawn, send re-enforcements" -> "fg djjack iu hadn, nwkh fdjndoscenjs"

plain text -> *cipher text*

The purpose of cryptography is to obscure data so as to make it unreadable without knowledge of the **key** - the method by which you can unscramble the cipher text back to plain text.

The Caesar Cipher

Used as long ago as 1st century BC the Roman would encrypt military messages with a code-wheel.

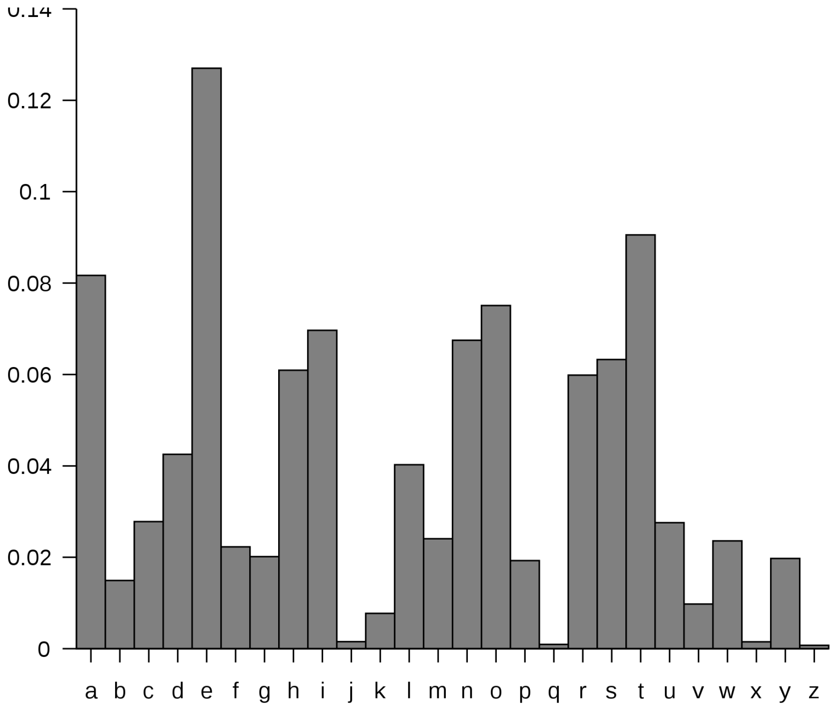
- The key is a number from 2 - 26 which tells you where to set the wheel and then it's just a case of translating letters to their cipher-text equivalent.
- In this case "**we attack at dawn**" would be rendered as "**jb fmmfdv fm cfjn**"
- In this case the **key is 6** (we slipped the wheel six slots from A-A)
- The ROT13 Internet forum cipher is just this.

The inherent problem with a Caesar cipher is letter-frequency analysis. To defeat this cipher without the key this is the **attack** we would use.

Defeating the Caesar Cipher

In English the letter E occurs most often followed by T, A, O, etc.

- The Vigenère cipher (AKA “la cifra indescifrable”) is an example of a Rotating Caesar Cipher.
- Extensively used by revolutionary French forces and then Napoleon (as well as during the American Civil War) it was thought to be uncrackable,
- Every letter you shift the code-wheel to a new position this breaking up the letter-frequency distribution,
- Typical keys are six (or eight) digits indicating a rotation of six positions before you return to the original position,
- Charles Babbage broke the cipher in 1854 but the British government persuaded him to keep it quiet!
- Each cipher-text is now six interleaved messages, all of which are susceptible to letter frequency analysis.



The Enigma machine - the last word in rotating code ciphers

If the vulnerability of the Vigenère cipher is the modest key length (typ. six characters) then the way to totally destroy any chance of using frequency analysis is to make the rotation cycle much bigger than the cipher text.

Show ’n’ Tell

The Enigma is just a bunch of wires, keys, bulbs and rotating wheels that change on every key-press.

- The current from each keypress travels through three or four rotors (depending on model), each wheel doing a letter-transposition.
- The “reflektor” send the current back through the wheels
- The patch-board allows another set of scrambling

This gives in the order of 26^{10} permutations of the machine = 141,167,095,653,376 (141 trillion) combinations. There are also more wheels to choose from on a daily basis so it’s actually much more than that.

Breaking the Enigma cipher

A code book was distributed by the Abwher every four weeks with the machine’s initial settings,

- So long as the wireless operators at each end of the link set their machines in the same configuration then plain text - cipher text - plain text works faultlessly,
- One Enigma weakness is the inability to encode a letter to itself
- Nineteen year-old conscripts often don’t follow procedure!

There has been much written about Bletchley Park and the electromechanical combination testing machines they built to speed up decrypting “X” traffic.

When properly used Enigma is almost unbreakable, even with modern, fast computers.

Symmetric Cryptography

A key is used to transform plain text into cipher text

- The key does not change and can reverse the process
- This developed into the “rotating Caesar Cipher”
- Eventually the peak was the Enigma machine (and others; Japanese “Purple” etc)
- Modern keys are 128 or more binary digits,
- Very fast to do using XOR gates (or similar) in hardware,
- Depending on application we may be using a **stream cipher** or a **block cipher**

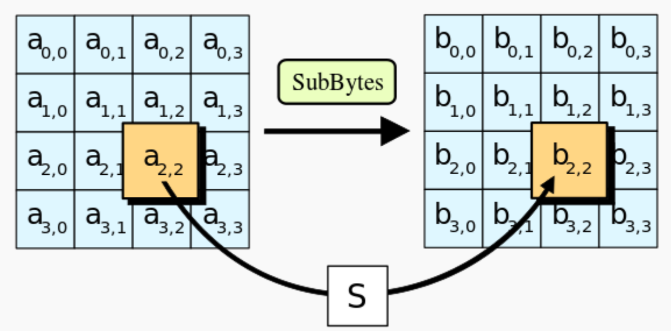
The takeaway is that when using a Symmetric Cipher the **encoding key is the same as the decoding key** - regardless of the cipher used.

Modern examples of Symmetric Ciphers

AES (Rijndael), Twofish, Blowfish, RC4 (Wifi’s original WEP stream cipher uses this), 3DES (“triple-DES”)

AES / Rijndael

- 128, 192 or 256-bit block cipher
- Although it requires more complicated operation that XOR’ing values it is very simple and is commonly implemented in hardware where needed.
- Attacks have been published that are computationally faster than a full brute force attack, though none as of 2013 are computationally feasible - what mathematicians describe as “non trivial” (sic)!


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
S
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

The `SubBytes` step, one of four stages in a round of AES

- AES does not rely on **security by obscurity** - the algorithm is open and understood; the security comes from the strength of the keying process, much like the Enigma.

The problem of Symmetric Ciphers

The problem with symmetric crypto is that you have to agree a key with the person you want to communicate securely with; if the bad guy is monitoring your network he can just observe the key and decode everything.

Wouldn’t work for PayPal, eBay, Amazon etc!

Alice, Bob & Eve are fictional actors in crypto scenarios; there was an early proposal for two-way symmetric crypto to avoid the problem of key exchange. Boxes & padlocks as an analogy are also involved…

- Alice locks her message with her key, sends it to Bob,
- Bob locks the received message with his key, returns it to Alice,
- Alice decrypts the message using her key; it’s still got Bob’s encryption, she returns it,
- Bob decrypts it, at no point did they need to exchange keys OR unencrypted data.

What’s the problem with this? ***What could Eve possibly do to intercept and read the message?***

Asymmetric / “Public Key” Cryptography

Again, a key is used to transform plain text into cipher text but crucially different keys are used; an **encrypting key** and a **decrypting key** which are entirely different and cannot be derived from each other.

The principle of most asymmetric ciphers is the “one-way-ness” of a mathematical function. This was discovered by Whitfield Diffie and Martin Hellman and independently by an unnamed researcher at GCHQ in the 70’s.

1. In the case of **RSA** (used very widely) the principle is that the product of two ***very large*** prime numbers is “none-trivial” to factor back to the original prime numbers.

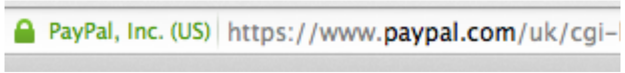
2. **Elliptic curve** is a plane curve over a finite field (rather than the real numbers) which consists of the points satisfying the equation

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

Selecting multiple points on any given curve does not allow you to derive the curve itself; this is the "elliptic curve discrete logarithm problem"

Asymmetric / “Public Key” Cryptography *cont.*

- Very slow to compute, not easily done in hardware - the keys are typically thousands of binary bits (4096 is becoming typical now).
- Only really used to securely exchange a symmetric key before a secure sessions starts;
- Web browsers come supplied with a **Certificate Store** that has all the public keys of the big web providers
- The other half; the **private keys** are stored securely at Mr. PayPal’s data centre (for example)
- With the public half of the key you can encrypt a message to the server that carries a symmetric key; then you can start communicating using that symmetric key that your web browser generated on the fly - this is referred to as an **ephemeral key** (or a **one-time-pad**)
- Neither the unencrypted key nor any unencrypted data passes the **m**an-**in**-**t**he-**m**iddle (the fictitious Eve)
- BUT, unlike the previous **MitM attack** you know for certain that you are dealing with Mr. PayPal because only their private key can decrypt what you send using the public key.



PayPal, Inc. (US) https://www.paypal.com/uk/cgi-

Protocols behind Public Key Crypto

SSL (or more accurately TLS nowadays) is the session protocol for starting a secure session.

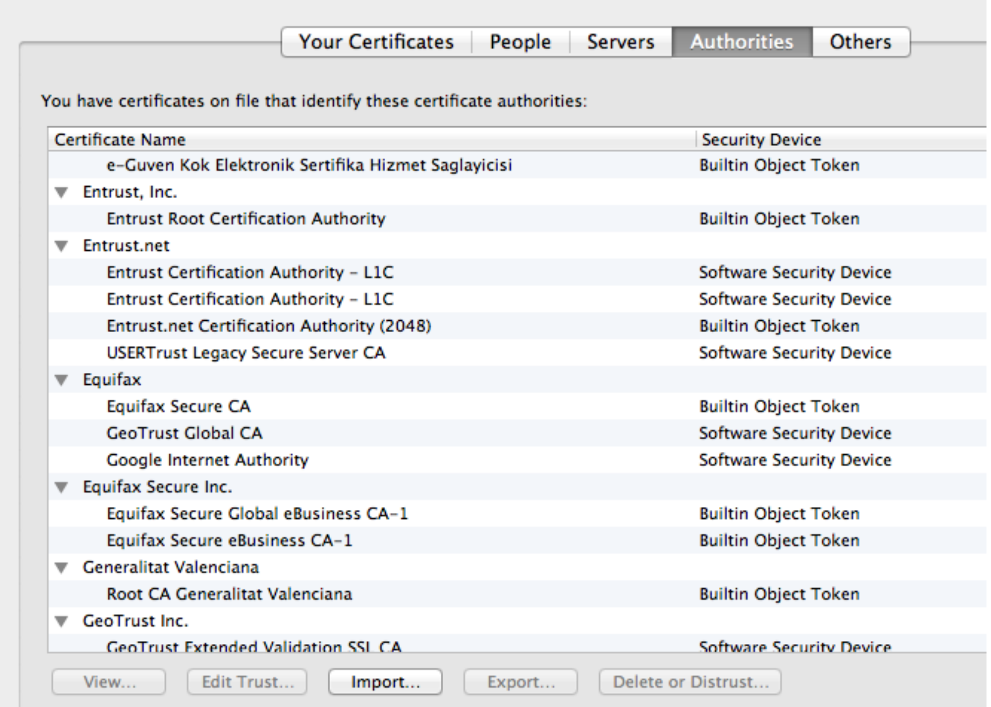
Once established via a public key there is a handshaking phase where the server and client decide which symmetric cipher will be used. The list provided by the web server (IIS, Apache etc) should be ordered with the strongest cipher first and the weakest last, maybe;

1. AES256
2. TwoFish
3. RC4
4. 3DES
5. DES

A modern browser and modern server will *hopefully* agree on a strong symmetric cipher from the **cipher suite** for the session.

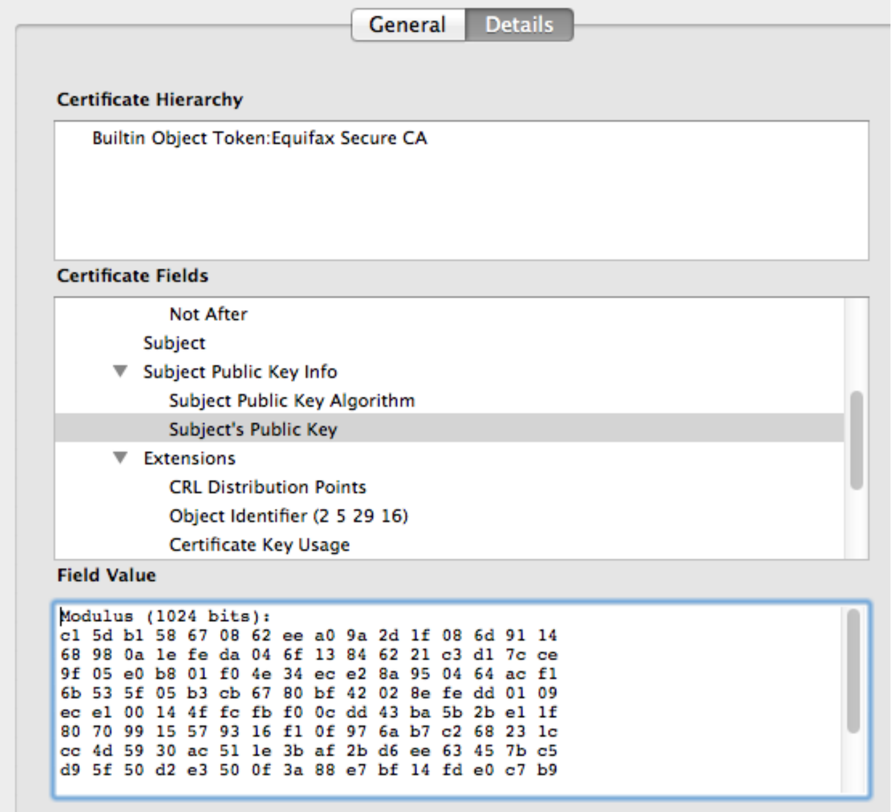
If you go browsing with Internet Explorer v.4 you my find you are less secure!

Public Key Cryptography - certificates



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
e-Guven Kok Elektronik Sertifika Hizmet Saglayicisi	Builtin Object Token
▼ Entrust, Inc.	
Entrust Root Certification Authority	Builtin Object Token
▼ Entrust.net	
Entrust Certification Authority - L1C	Software Security Device
Entrust Certification Authority - L1C	Software Security Device
Entrust.net Certification Authority (2048)	Builtin Object Token
USERTrust Legacy Secure Server CA	Software Security Device
▼ Equifax	
Equifax Secure CA	Builtin Object Token
GeoTrust Global CA	Software Security Device
Google Internet Authority	Software Security Device
▼ Equifax Secure Inc.	
Equifax Secure Global eBusiness CA-1	Builtin Object Token
Equifax Secure eBusiness CA-1	Builtin Object Token
▼ Generalitat Valenciana	
Root CA Generalitat Valenciana	Builtin Object Token
▼ GeoTrust Inc.	
GeoTrust Extended Validation SSL CA	Software Security Device



Certificate Hierarchy

Builtin Object Token:Equifax Secure CA

Certificate Fields

- Not After
- Subject
- ▼ Subject Public Key Info
  - Subject Public Key Algorithm
  - Subject's Public Key
- ▼ Extensions
  - CRL Distribution Points
  - Object Identifier (2 5 29 16)
  - Certificate Key Usage

Field Value

```
Modulus (1024 bits):
c1 5d b1 58 67 08 62 ee a0 9a 2d 1f 08 6d 91 14
68 98 0a 1e fe da 04 6f 13 84 62 21 c3 d1 7c ce
9f 05 e0 b8 01 f0 4e 34 ec e2 8a 95 04 64 ac f1
6b 53 5f 05 b3 cb 67 80 bf 42 02 8e fe dd 01 09
ec e1 00 14 4f fc fb f0 0c dd 43 ba 5b 2b e1 1f
80 70 99 15 57 93 16 f1 0f 97 6a b7 c2 68 23 1c
cc 4d 59 30 ac 51 1e 3b af 2b d6 ee 63 45 7b c5
d9 5f 50 d2 e3 50 0f 3a 88 e7 bf 14 fd e0 c7 b9
```

This is one of the certificates in Firefox’s certificate store for Equifax – a provider of certificate trust.

Cryptography, hashing functions

A cryptographic hash function allows one to easily verify that some input data matches a stored hash value, but makes it hard to reconstruct the data from the hash alone. They are referred to as “one-way-functions”

Common hashing algorithms are;

- MD5 – now considered insecure
- SHA1 – starting to show it’s age, 160 bits
- SHA256 – now the preferred one at 256 bits

A website’s certificate can be used to “sign” a file or other block of data to prove where it has come from since the hash function is not practically undoable. Message integrity is another way hashes are used.

The *chain of trust* of web certificates may well depend on several certificates in the chain, each *signed* by a more senior **certificate authority**.

Common crypto attacks and cipher strength

You often read in the *fashionable* tech press (Wired, The Register, Gizmodo etc) that ***xyz-cipher*** has been "cracked". More often than not the crypto is strong, but the **implementation is bad** OR the **breach is unrealistic**;

- Downgrade attack - **Logjam** was a famous downgrade attack where a man-in-the-middle was able to strip out the securest ciphers from the server's response before the TLS session started causing all web traffic to be weakly encrypted.
- Side-channel attack - by timing the egress of IP packets from a machine during encrypted credit card number entry the credit card number may be determined.
- Poor random numbers - the European ATM protocol **EMV** depends on good-quality random numbers at the cash machine end; a compromised ATM allows an attacker to predict the symmetric key used for the session.
- re-use of temporary keys - the older WiFi **WEP protocol** re-used temporary keys many times allowing attackers a foothold with differential data.
- Intentional weakening of a cypher; **CSS** used in DVD uses a 40-bit cipher (rather than 64-bit) due to **military-export restraints**. By the early noughties home-PCs were fast enough to crack this.

Secure file transport and whole disk encryption

Starting with Secure FTP there are many well-secured methods used in broadcast file delivery;

- Signiant
- Aspera
- FileCatalyst

These all use SSL/TLS to start the session (public key) and AES (symmetric crypto) to encrypt the data in transit.

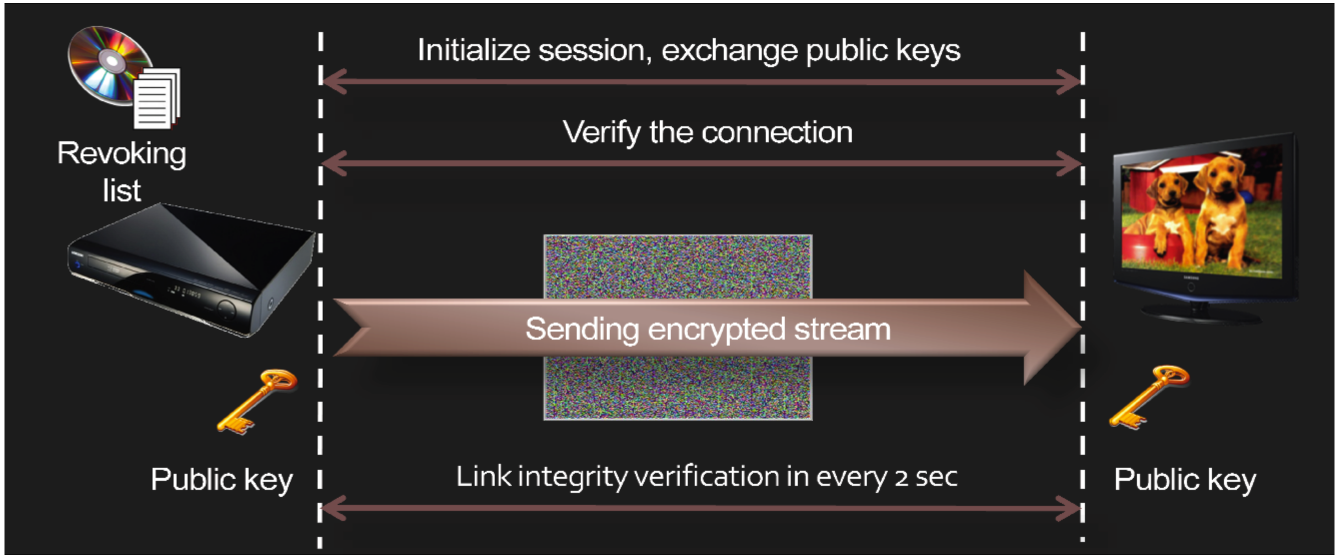
On a local machine whole-disk encryption may use;

- Truecrypt - cross platform but now deprecated (unfortunately!)
- Bitlocker on Windows
- FileVault on Mac

Why can't I convert my Blue Ray disk to HD-SDi and capture it?

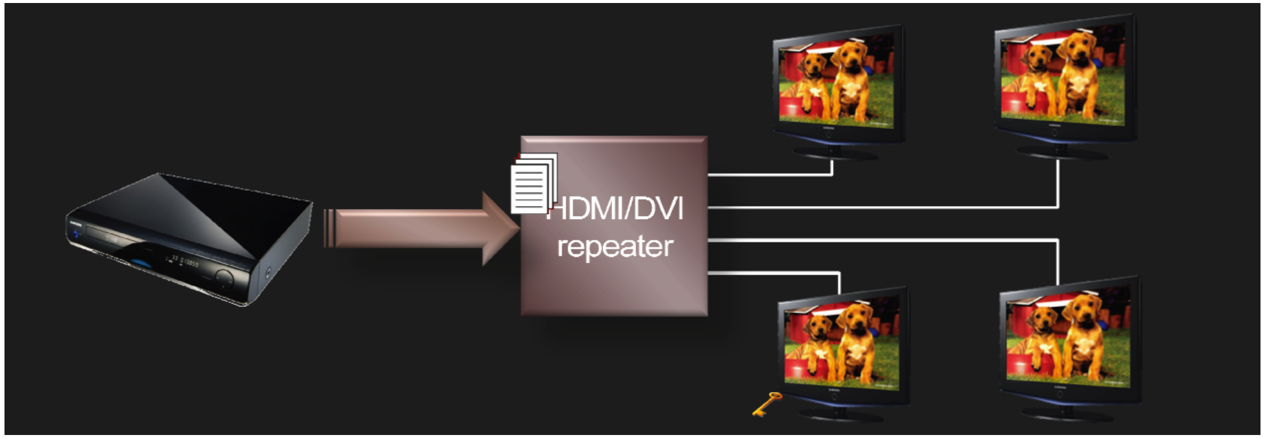
HDCP – ***High Definition Content Protection*** system.

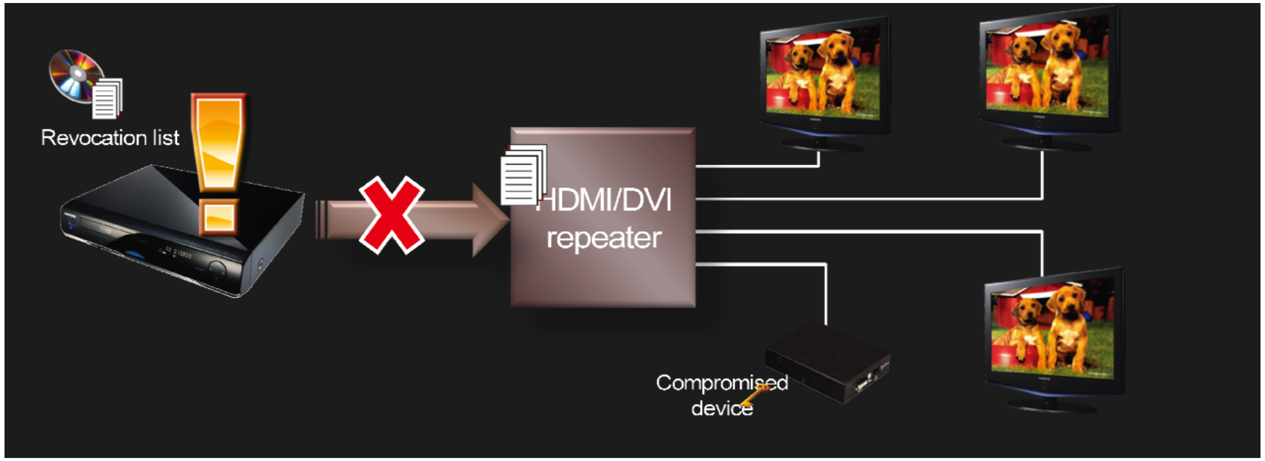
- Industrial strength public/private key cryptography
- Each player has ***device keys*** and each disk ***volume keys***
- These are combined and used to decrypt the content using a symmetric stream cipher
- Hollywood has the ability to disable a device (Sony BD player, for example) on new releases by use of revocation lists in new content.

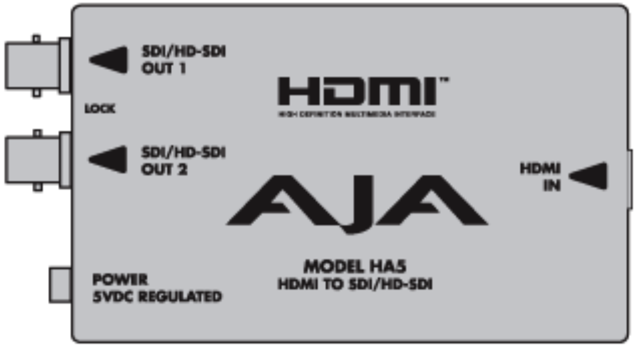

Revoking list
Initialize session, exchange public keys
Verify the connection
Sending encrypted stream
Public key
Link integrity verification in every 2 sec
Public key

HDCP cont.

- If there are multiple *"sinks"* then the key exchange has to happen several times.
- The *"repeater"* (HDMI distribution amplifier or router) has to manage/ arbitrate this process.
- If a *"source"* is updated by the disk *"revocation list"* then a sink can be disabled permanently.
- **Only one** compromised sink will spoil the process for all.







There is **no way** a manufacturer can remove HDCP encryption and expect their product to work for more than a few weeks – the Hollywood Alliance revokes keys when it discovers this! New content won't play ***AND*** old devices will not handshake if their revocation list gets updated.

Hollywood’s long reach!

In early 2010 it was discovered that Cyberlink PowerDVD playback software left it’s ***device key*** present in memory whilst playing back HD-DVDs and BluRay disks. By using the Windows Debugger (in the Visual Basic IDE) it’s possible to freeze the running executable and step up through memory to find the key.

- With the ***title ID*** you can now select the correct ***volume key*** and combined with the discovered ***device key*** you can decrypt the symmetric stream cipher used to encode the video and audio data.
- Very soon PasteBin and other hacker websites were hosting usable keys for all the blockbusters
- Very soon after that the Hollywood Alliance revoked PowerDVD’s device key!
- So all new titles become unplayable on revoked software - no matching ***volume key*** is included.
- All new titles have a ***revocation list*** containing the offending ***device keys*** which other devices (monitors, TV etc) are obliged to ignore if they want to keep on the right side of the MPAA.

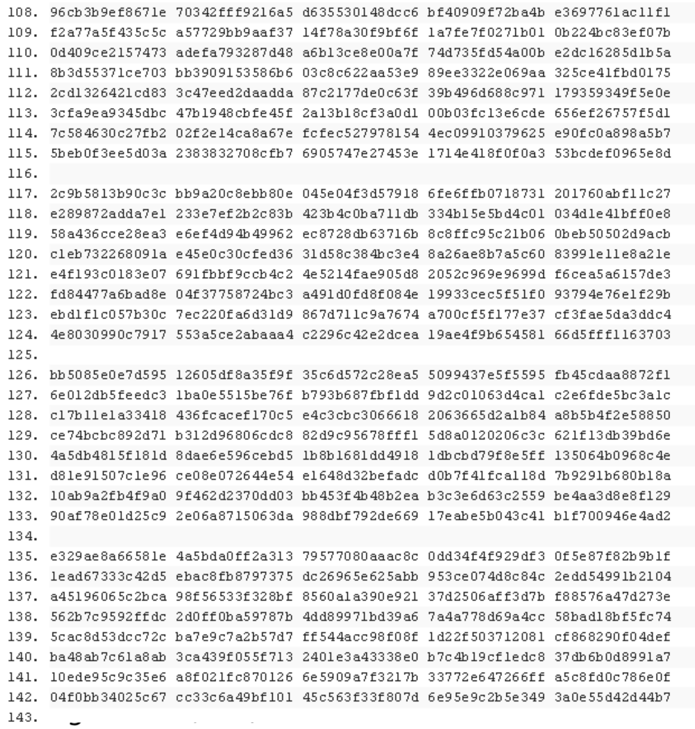
Hollywood done goofed up!

With such a robust method for protecting content you'd think they would have avoided back-doors!

In September 2010 the master ***volume-device key*** pair was leaked and now it entirely possible to decrypt any HDCP encoded material. This principally opened the floodgate for every sell-through disk to be easily ripped and made available on Bittorent sites.

This key was only intended for internal engineering purposes and to revoke it would cause an awful lot of devices to become useless.

Thankfully what little use we make of HDMI in pro applications is not HDCP encumbered.

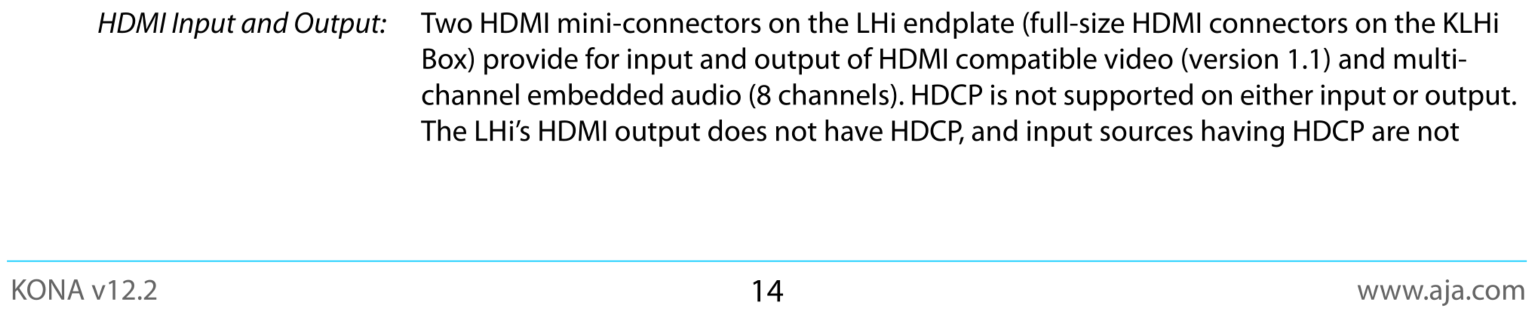


```
96cb3b9ef8671e 70342fff9216a5 d635530148dcc6 bf40909f72ba4b e369776lac11f1
f2a77a5f435c5c a57729bb9aaf37 14f78a30f9bf6f 1a7fe7f0271b01 0b224bc83ef07b
0d409ce2157473 adefa793287d48 a6b13ce8e00a7f 74d735fd54a00b e2dc16285d1b5a
8b3d55371ce703 bb3909153586b6 03c8c622aa53e9 89ee3322e069aa 325ce41fbd0175
2cd1326421cd83 3c47eed2daadda 87c2177de0c63f 39b496d688c971 179359349f5e0e
3cfa9ea9345dbc 47b1948cbfe45f 2a13b18cf3a0d1 00b03fc13e6cde 656ef26757f5d1
7c584630c27fb2 02f2e14ca8a67e fcfec527978154 4ec09910379625 e90fc0a898a5b7
5beb0f3ee5d03a 2383832708cfb7 6905747e27453e 1714e418f0f0a3 53bcdef0965e8d

2c9b5813b90c3c bb9a20c8ebb80e 045e04f3d57918 6fe6ffb0718731 201760abf11c27
e289872adda7e1 233e7ef2b2c83b 423b4c0ba711db 334b15e5bd4c01 034d1e41bff0e8
58a436cce28ea3 e6ef4d94b49962 ec8728db63716b 8c8ffc95c21b06 0beb50502d9acb
c1eb73226809la e45e0c30cfed36 31d58c384bc3e4 8a26ae8b7a5c60 83991e11e8a21e
e4f193c0183e07 691fbbf9ccb4c2 4e5214fae905d8 2052c969e9699d f6cea5a6157de3
fd84477a6bad8e 04f37758724bc3 a491d0fd8f084e 19933cec5f51f0 93794e76e1f29b
ebd1f1c057b30c 7ec220fa6d31d9 867d711c9a7674 a700cf5f177e37 cf3fae5da3ddc4
4e8030990c7917 553a5ce2abaaa4 c2296c42e2dcea 19ae4f9b654581 66d5fff1163703

bb5085e0e7d595 12605df8a35f9f 35c6d572c28ea5 5099437e5f5595 fb45cdaa8872f1
6e012db5feedc3 1ba0e5515be76f b793b687fbf1dd 9d2c01063d4ca1 c2e6fde5bc3a1c
c17b11e1a33418 436fcacef170c5 e4c3cbc3066618 2063665d2a1b84 a8b5b4f2e58850
ce74bcbc892d71 b312d96806cdc8 82d9c95678fff1 5d8a0120206c3c 621f13db39bd6e
4a5db4815f181d 8dae6e596cebd5 1b8b1681dd4918 1dbcbd79f8e5ff 135064b0968c4e
d81e91507c1e96 ce08e072644e54 e1648d32befadc d0b7f41fca118d 7b9291b680b18a
10ab9a2fb4f9a0 9f462d2370dd03 bb453f4b48b2ea b3c3e6d63c2559 be4aa3d8e8f129
90af78e01d25c9 2e06a8715063da 988dbf792de669 17eabe5b043c41 b1f700946e4ad2

e329ae8a66581e 4a5bda0ff2a313 79577080aaac8c 0dd34f4f929df3 0f5e87f82b9b1f
1ead67333c42d5 ebac8fb8797375 dc26965e625abb 953ce074d8c84c 2edd54991b2104
a45196065c2bca 98f56533f328bf 8560a1a390e921 37d2506aff3d7b f88576a47d273e
562b7c9592ffdc 2d0ff0ba59787b 4dd89971bd39a6 7a4a778d69a4cc 58bad18bf5fc74
5cac8d53dcc72c ba7e9c7a2b57d7 ff544acc98f08f 1d22f503712081 cf868290f04def
ba48ab7c61a8ab 3ca439f055f713 2401e3a43338e0 b7c4b19cf1edc8 37db6b0d8991a7
10ede95c9c35e6 a8f021fc870126 6e5909a7f3217b 33772e647266ff a5c8fd0c786e0f
04f0bb34025c67 cc33c6a49bf101 45c563f33f807d 6e95e9c2b5e349 3a0e55d42d44b7
```



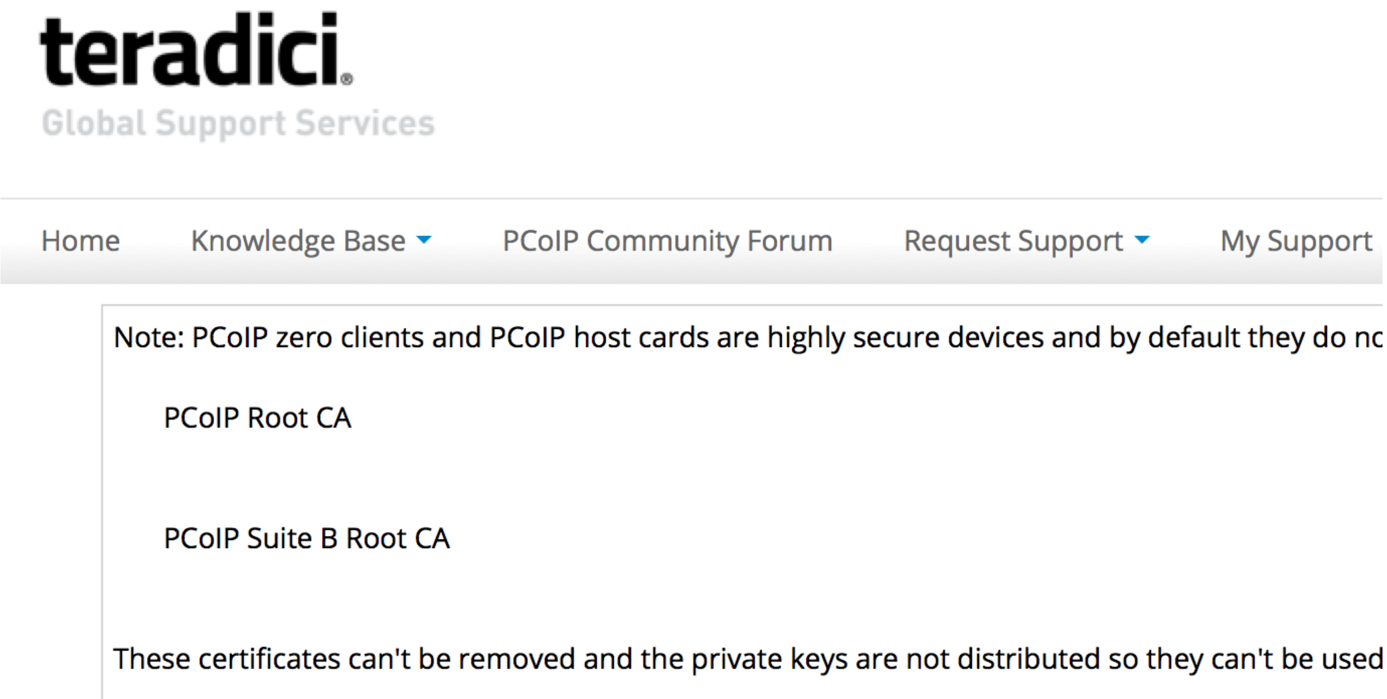
HDMI Input and Output: Two HDMI mini-connectors on the LHi endplate (full-size HDMI connectors on the KLHi Box) provide for input and output of HDMI compatible video (version 1.1) and multi-channel embedded audio (8 channels). HDCP is not supported on either input or output. The LHi's HDMI output does not have HDCP, and input sources having HDCP are not

KONA v12.2 14 www.aja.com

from the AJA Kona manual

PCoIP – security of KVM extenders

We have several-hundred seats of Amulet in a mix of VFX, broadcast, audio, post, education and in fact the case of facilities that come under Marvel / MPAA / Disney a Teradici solution is pretty mandated as they are the only one who do encryption properly (other KVM-over-IP manufacturers use symmetric encryption on the wire but they do the key-exchange in the clear; Amulet do proper asymmetric public-key crypto).



Note: PCoIP zero clients and PCoIP host cards are highly secure devices and by default they do nc

PCoIP Root CA

PCoIP Suite B Root CA

These certificates can't be removed and the private keys are not distributed so they can't be used

PCoIP – security of KVM extenders *cont.*

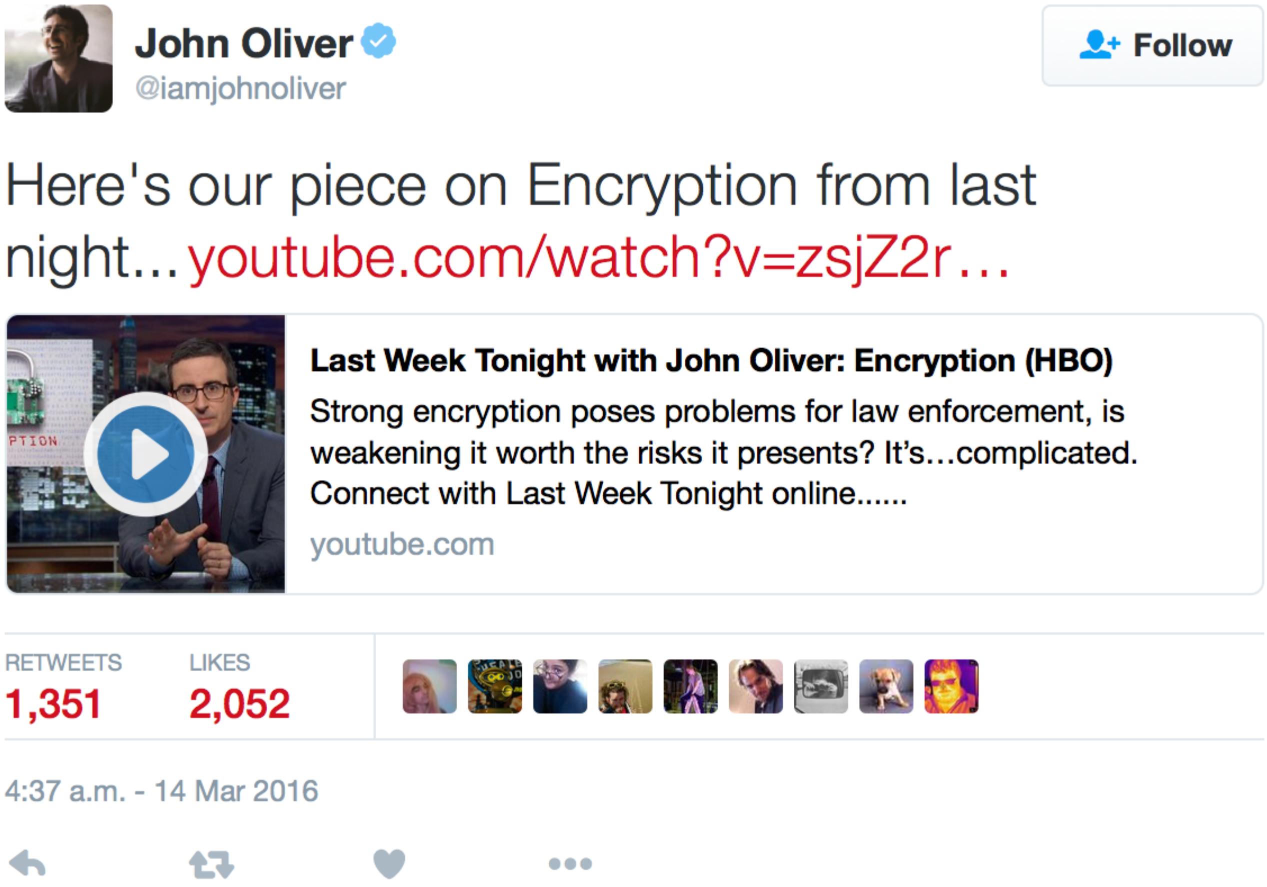
So because every PCoIP session involves an asymmetric (public key) exchange before the bulk data is encrypted using a symmetric 256-bit AES cipher the following are factors;

1. You can’t have one desktop feeding multiple stations,
2. You can’t do instant switching between hosts.

The other manufacturers claim these as features for their systems but bear in mind that what it tell us is that they aren’t doing their crypto in a secure fashion.

Final Thought

If the government get to specify who can use what kind of crypto and when - what will it mean for the Film & TV industry?



John Oliver
@iamjohnoliver

Here's our piece on Encryption from last night... youtube.com/watch?v=zsjZ2r…

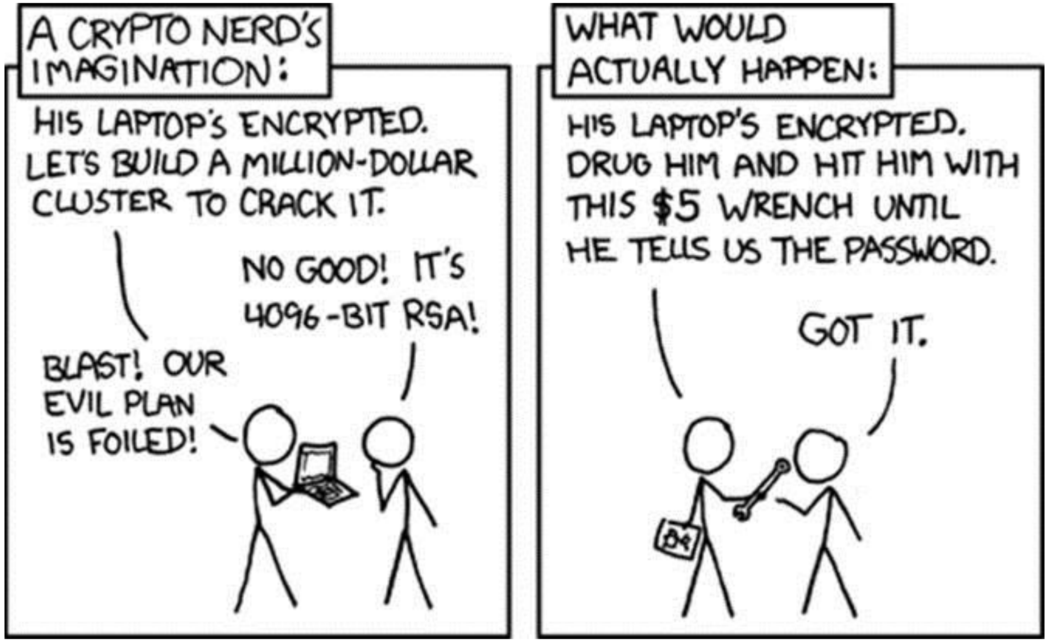
Last Week Tonight with John Oliver: Encryption (HBO)
Strong encryption poses problems for law enforcement, is weakening it worth the risks it presents? It's…complicated. Connect with Last Week Tonight online......
youtube.com

RETWEETS 1,351
LIKES 2,052

4:37 a.m. - 14 Mar 2016

Recommended Reading

- **The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography** by *Simon Singh*
- **Security Engineering** by *Ross Anderson* `http://www.cl.cam.ac.uk/~rja14/book.html`
- *Steve Gibson*'s **Security Now** podcast (ep. 31– 37 are a great crypto primer) `https://www.grc.com/securitynow.htm`
- **The Secrets of Station** X by *Michael Smith*.
- **Cryptography Engineering - Design Principles and Practical Applications** by *Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno* `https://www.schneier.com/books/cryptography_engineering`
- `http://philtechnicalblog.blogspot.co.uk/search/label/cryptography`


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

most of my crypto knowledge comes from XKCD