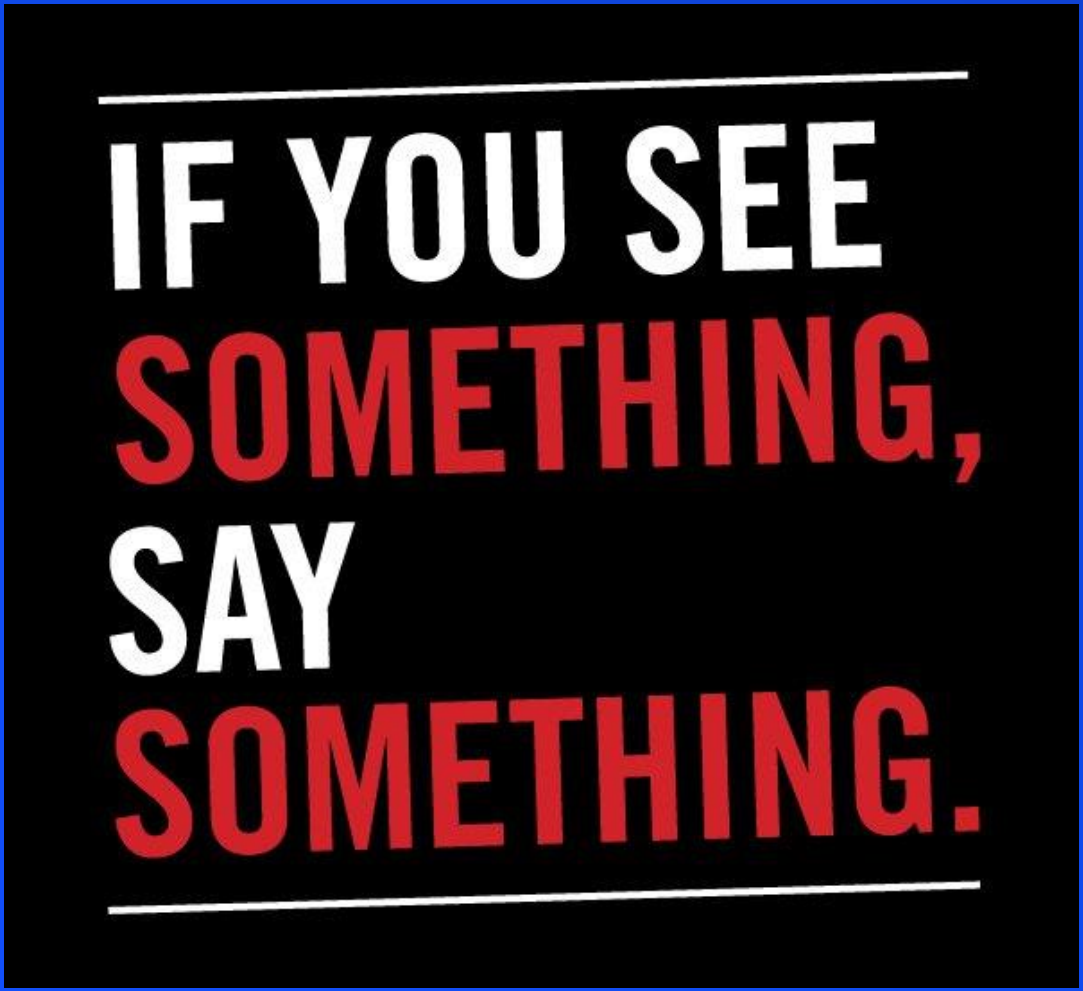
Crypto Won’t Save You Either

Peter Gutmann

University of Auckland

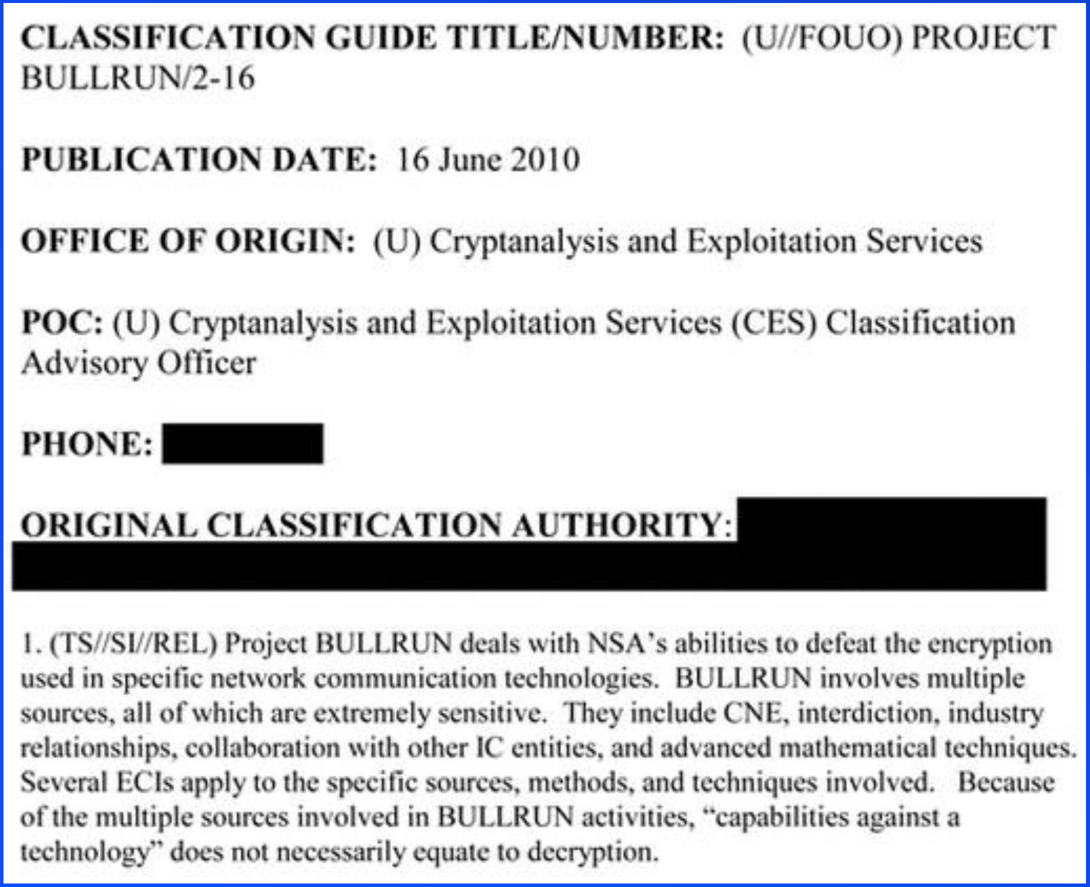
Sound Advice from the USG



IF YOU SEE SOMETHING, SAY SOMETHING.

Saw Something, Said Something

Saw Something, Said Something (ctd)



CLASSIFICATION GUIDE TITLE/NUMBER: (U//FOUO) PROJECT BULLRUN/2-16

PUBLICATION DATE: 16 June 2010

OFFICE OF ORIGIN: (U) Cryptanalysis and Exploitation Services

POC: (U) Cryptanalysis and Exploitation Services (CES) Classification Advisory Officer

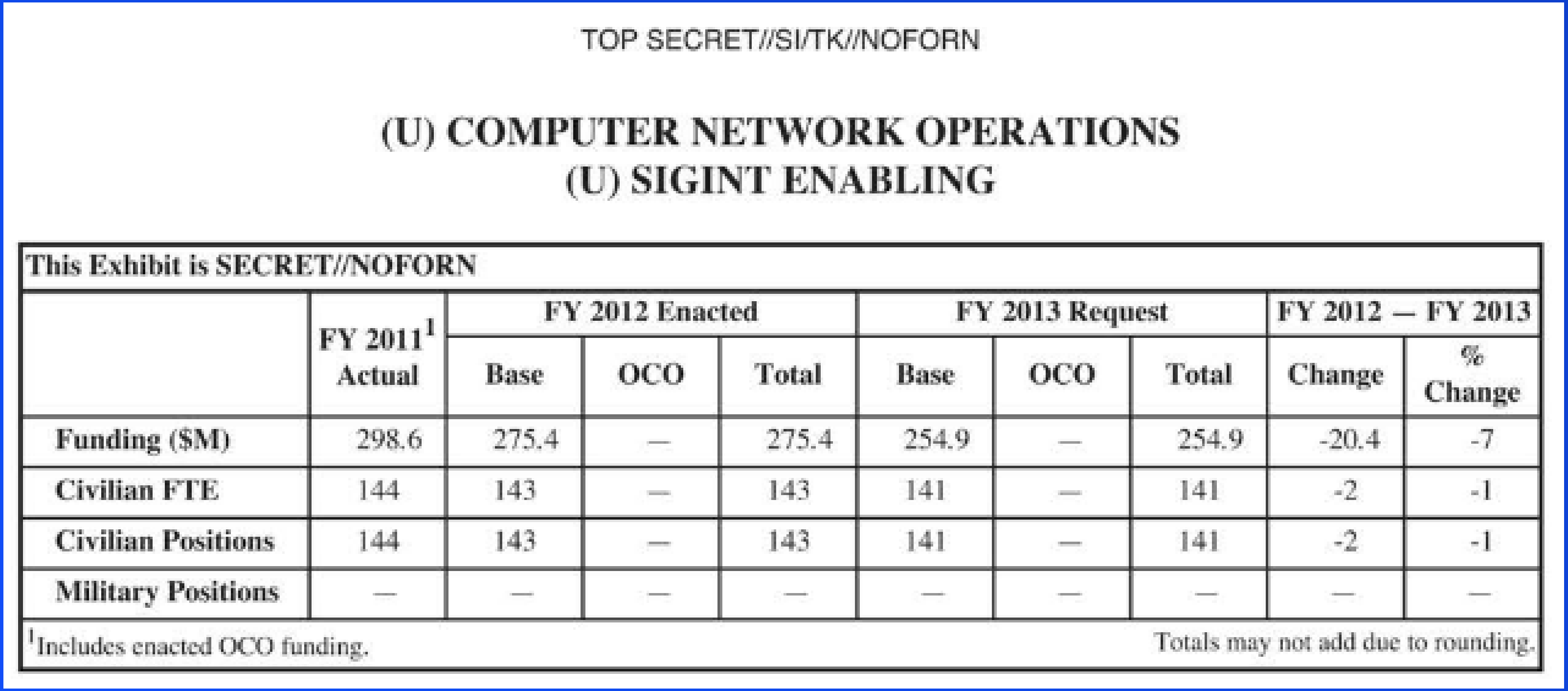
PHONE: [redacted]

ORIGINAL CLASSIFICATION AUTHORITY: [redacted]

1. (TS//SI//REL) Project BULLRUN deals with NSA's abilities to defeat the encryption used in specific network communication technologies. BULLRUN involves multiple sources, all of which are extremely sensitive. They include CNE, interdiction, industry relationships, collaboration with other IC entities, and advanced mathematical techniques. Several ECIs apply to the specific sources, methods, and techniques involved. Because of the multiple sources involved in BULLRUN activities, "capabilities against a technology" does not necessarily equate to decryption.

You're not paranoid, they really are out to get you

BULLRUN



TOP SECRET//SI/TK//NOFORN

(U) COMPUTER NETWORK OPERATIONS
(U) SIGINT ENABLING

This Exhibit is SECRET//NOFORN

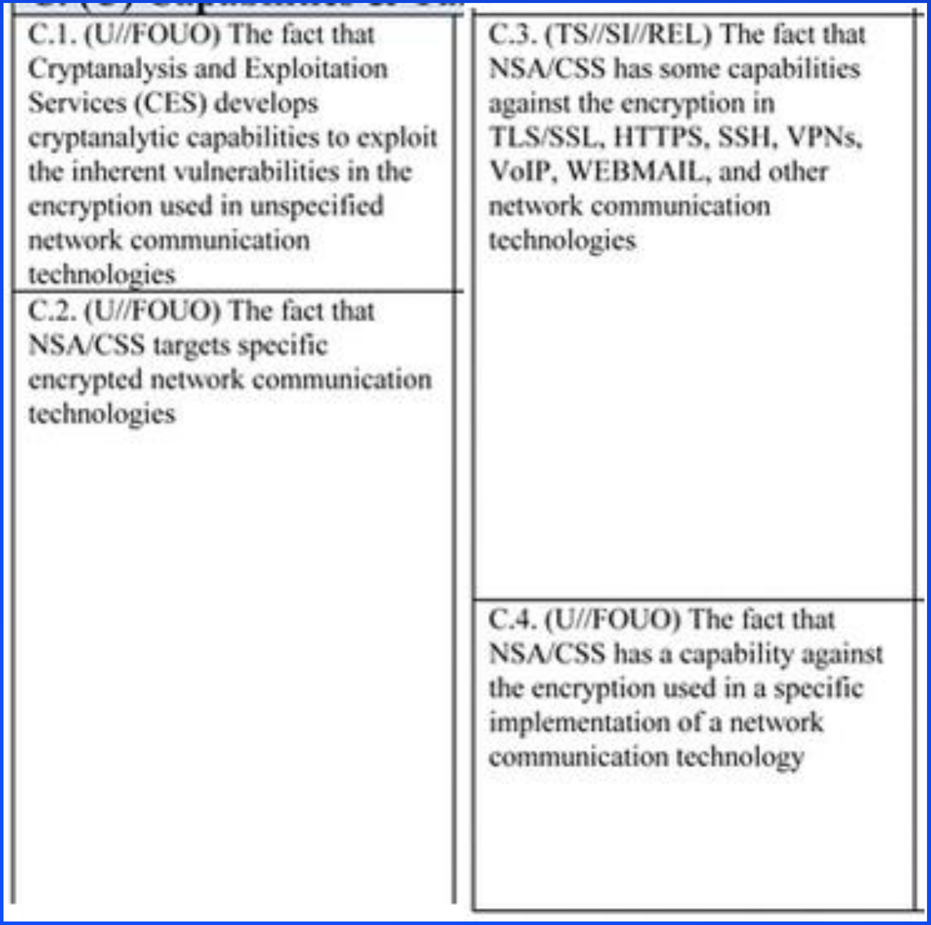
	FY 2011[1] Actual	FY 2012 Enacted			FY 2013 Request			FY 2012 — FY 2013	
		Base	OCO	Total	Base	OCO	Total	Change	% Change
Funding ($M)	298.6	275.4	—	275.4	254.9	—	254.9	-20.4	-7
Civilian FTE	144	143	—	143	141	—	141	-2	-1
Civilian Positions	144	143	—	143	141	—	141	-2	-1
Military Positions	—	—	—	—	—	—	—	—	—

[1] Includes enacted OCO funding.

Totals may not add due to rounding.

Funded to the tune of $250-300M/year

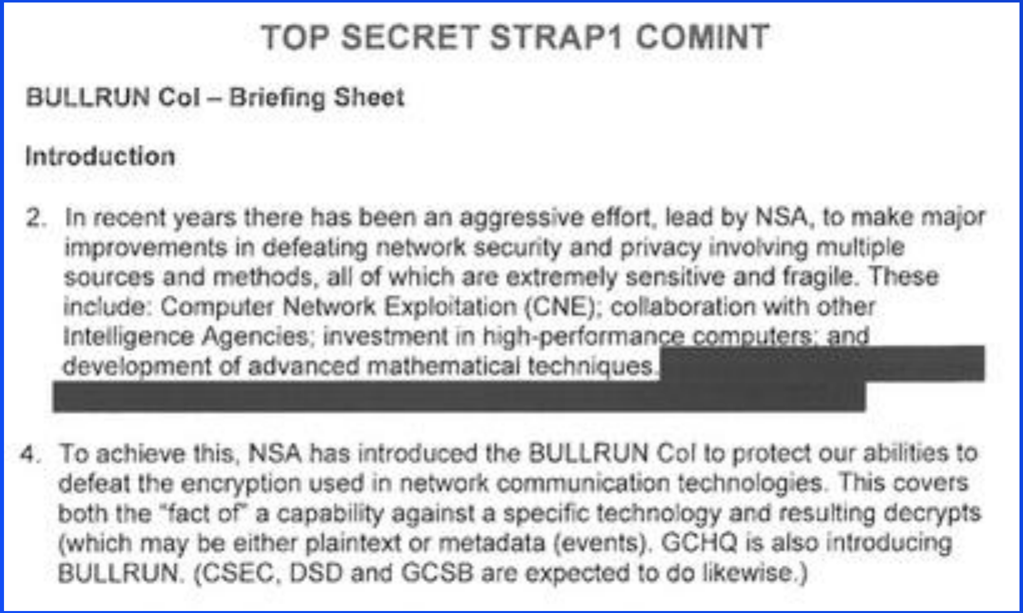
BULLRUN (ctd)



C.1. (U//FOUO) The fact that Cryptanalysis and Exploitation Services (CES) develops cryptanalytic capabilities to exploit the inherent vulnerabilities in the encryption used in unspecified network communication technologies	C.3. (TS//SI//REL) The fact that NSA/CSS has some capabilities against the encryption in TLS/SSL, HTTPS, SSH, VPNs, VoIP, WEBMAIL, and other network communication technologies
C.2. (U//FOUO) The fact that NSA/CSS targets specific encrypted network communication technologies	C.4. (U//FOUO) The fact that NSA/CSS has a capability against the encryption used in a specific implementation of a network communication technology

"capabilities against TLS/SSL, HTTPS, SSH, VPNs, VoIP, webmail, ..."

BULLRUN (ctd)



TOP SECRET STRAP1 COMINT

BULLRUN CoI – Briefing Sheet

Introduction

2. In recent years there has been an aggressive effort, lead by NSA, to make major improvements in defeating network security and privacy involving multiple sources and methods, all of which are extremely sensitive and fragile. These include: Computer Network Exploitation (CNE); collaboration with other Intelligence Agencies; investment in high-performance computers; and development of advanced mathematical techniques. [redacted]

4. To achieve this, NSA has introduced the BULLRUN CoI to protect our abilities to defeat the encryption used in network communication technologies. This covers both the "fact of" a capability against a specific technology and resulting decrypts (which may be either plaintext or metadata (events). GCHQ is also introducing BULLRUN. (CSEC, DSD and GCSB are expected to do likewise.)

"aggressive effort to defeat network security and privacy"

"defeat the encryption used in network communication technologies"

What’s that NSAie? Crypto’s fallen in the well?

I Know, Bigger Keys!

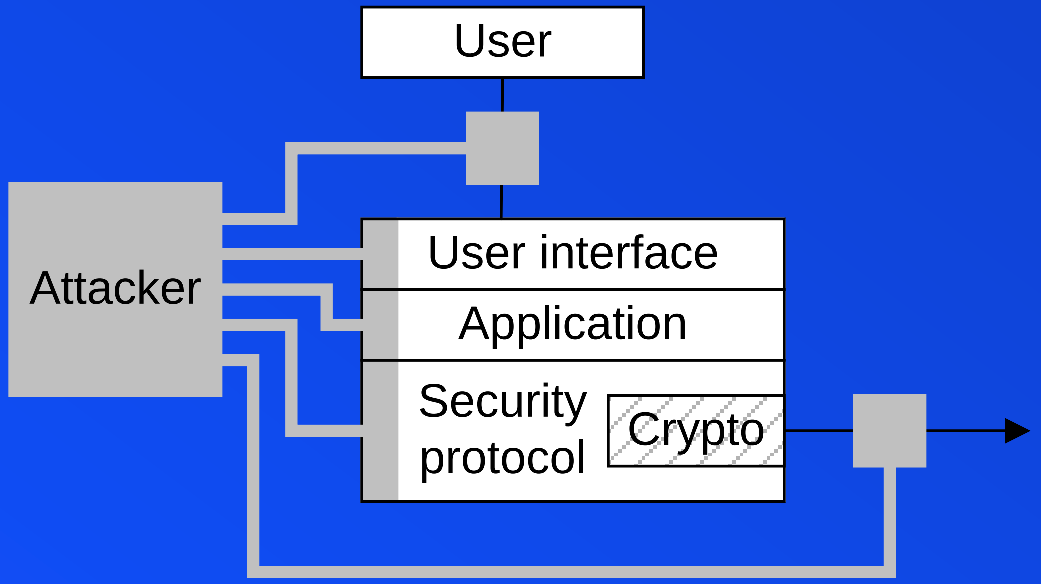
We need to get bigger keys. BIG F**ING KEYS!

— "Deep Impact", 1992

Quick, do something!

Cue the stannomillinery

Crypto Won't Save You



Shamir's Law: Crypto is bypassed, not penetrated

Cryptography is usually bypassed. I am not aware of any major world-class security system employing cryptography in which the hackers penetrated the system by actually going through the cryptanalysis […] usually there are much simpler ways of penetrating the security system — Adi Shamir

Example: Games Consoles

All of the major consoles use fairly extensive amounts of sophisticated cryptography

- PS3
- Wii
- Xbox
- Xbox 360

Example: Games Consoles (ctd)

Measures include

- Signed executables
- Encrypted storage
- Full-media encryption and signing
- Memory encryption and integrity-protection
- On-die key storage and/or use of security coprocessors
  - If you asked someone a decade ago what this was describing, they'd have guessed an NSA-designed crypto box

All of them have been hacked

- In none of the cases was it necessary to break the cryptography

Crypto Won’t Save You

Amazon Kindle 2

- All binaries signed with a 1024-bit RSA key
- Jailbreakers replaced it with their own one
- Later versions of the Kindle were similarly jailbroken without breaking the crypto

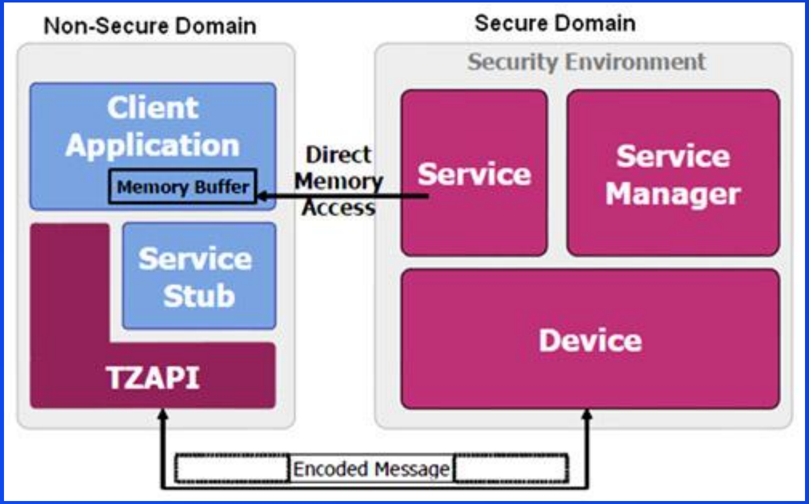
HTC Thunderbolt

- Signed binaries
- Signed kernel
- Signed system-recovery/restart code
- Remove the signature-checking code

Crypto Won't Save You (ctd)

Motorola cellphones

- Careful chaining of hashes, MACs (keyed hashes), and digital signatures
- Ignore the crypto and target the ARM TrustZone hardware-enforced security system
- "It's secure, because we say it is!"



- Find exploit inside the trusted, secure kernel and attack the untrusted code from inside the trusted kernel
  - Bootloader code was (apparently) quite good, it was the trusted security kernel that was insecure

Crypto Won’t Save You (ctd)

Samsung Galaxy

- Firmware signed with 2048-bit RSA key
  - Round up twice the usual number of key bits!
- Modify firmware metadata to load it over the top of the signature-checking code

Nikon Cameras

- Sign images using a 1024-bit RSA key
- Signature encoded in photo EXIF data
- Signing key encoded in camera firmware…

Crypto Won’t Save You (ctd)

Canon Cameras

- Authenticate images using HMAC (keyed hash function)
- HMAC is symmetric: Verifier needs to know the key as well
- Shared HMAC key encoded in camera firmware…

Airport Express

- Signs data with a 2048-bit RSA key
- Recover the private key from the firmware image

Crypto Won’t Save You (ctd)

Diaspora

- Privacy-aware alternative to Facebook
- Replace the victim’s public key with your own one
- You can now MITM all of the victim’s messages

Google Chromecast

- Carefully verified signed image on loading
- Ignored the return value of the signature-checking function

Crypto Won’t Save You (ctd)

Google TV

- Range of devices from various manufacturers
- Exploit inadvertently-enabled debug modes
- Use improper path validation to run unapproved binaries
- Remap NAND flash controller registers to allow kernel memory overwrite
- Desolder encrypted SSD and replace with unencrypted one
- Usual plethora of Linux kernel bugs and application-level errors

Crypto Won’t Save You (ctd)

Android code signing

- APK = JAR = Zip file
- Signed using specially-named files included in the Zip archive (MANIFEST.MF, CERT.SF, CERT.RSA)
- Use custom archive tool to create Zip file with duplicate filenames
- Verification is done using a Java hashmap
  - Duplicate entries are overwritten
- Installation is done via C code
  - Duplicate entries are processed on the assumption that they’ve been sig-checked

Crypto Won’t Save You (ctd)

iPhone/iPad/iOS

- Lots of security measures, too many to cover here

Bypasses include

- Inject executable code as data pages
  - Data isn’t code so it’s not signature-checked
- Exploit debugging facilities present in signed OS components
- Use ROP to synthesise exploits from existing signed code fragments
- …

Crypto Won’t Save You (ctd)

Windows RT UEFI

- Exploit privilege escalation vulnerability in the RT kernel to bypass signing

Windows 8 UEFI

- Patch SPI flash memory holding UEFI firmware to skip the signature-check
- Clear flags in system NVRAM to disable signature checks

Crypto Won’t Save You (ctd)

CCC 2011 Badge

- Used Corrected Block TEA/XXTEA block cipher with 128-bit key
- Various exploits that all bypassed the need to deal with XXTEA
- Eventually, loaded custom code to extract the 128-bit key

It’s probably at least some sort of sign of the end times when your conference badge has a rootkit

Crypto Won’t Save You (ctd)

Xbox (earlier attack)

- Data moving over high-speed internal buses was deemed to be secure
- HyperTransport bus analysers existed only in a few semiconductor manufacturer labs

LVDS signalling looks a lot like HT signalling

- Use an LVDS transceiver to decode HT signalling

Standard FPGA’s aren’t fast enough to process the data

- Hand-optimise paths through the FPGA’s switching fabric
- Clock data onto four phases of a quarter-speed clock
  - 8-bit stream → 32-bit stream at ¼ speed
- Overclock the FPGA

Crypto Won’t Save You (ctd)

Xbox (later attacks)

- Force the CPU to boot off external ROM rather than secure internal ROM
  - Standard smart-card hacker’s trick
- Exploit architectural quirks in the CPU
  - Microsoft developed with AMD CPUs but shipped with an Intel CPU
- Exploit backwards-compatibility support in the CPU for bugs dating back to the 80286
- Exploit the fact that font files (TTFs) were never verified
  - Use doctored fonts to leverage a vulnerability in the Xbox font handler

Crypto Won’t Save You (ctd)

PS3

- Variant of the first Xbox attack
- Don’t try and pull data off the bus, just glitch it
- Processor now has an incorrect view of what’s stored in memory
  - Data in cache doesn’t match what’s actually in memory

Xbox 360

- Another glitch attack
- Ensure that a hash comparison always returns a hash-matched result

Crypto Won’t Save You (ctd)

Jailbreakers are rediscovering 15-20 year old smart card attacks

I never met a smart-card I couldn’t glitch
— European smart card hacker

Example: Clock glitches

- Send multiple clock pulses in the time interval when a single pulse should occur
- Fast-reacting parts of the CPU like the program counter respond
- Slower-reacting parts of the CPU like the ALU don’t have time
- Skip instructions, e.g. ones that perform access-control checks

Some Metrics…

How unnecessary is it to attack the crypto?

Geer’s Law:

> Any security technology whose effectiveness can’t be empirically determined is indistinguishable from blind luck
>
> — Dan Geer

Some Metrics… (ctd)

Large-scale experiment carried out by a who’s-who of companies

- Amazon
- Apple
- Dell
- eBay
- HP
- HSBC
- LinkedIn
- Paypal
- Twitter

Some Metrics… (ctd)

In late 2012, researchers noticed that these organisations, and many others, were using toy keys for DKIM signing

- 12,000 organisations
- 4,000 were using keys so weak that an individual attacker could have broken them

If this crypto was so weak, why didn’t anyone attack it?

- It wasn’t necessary

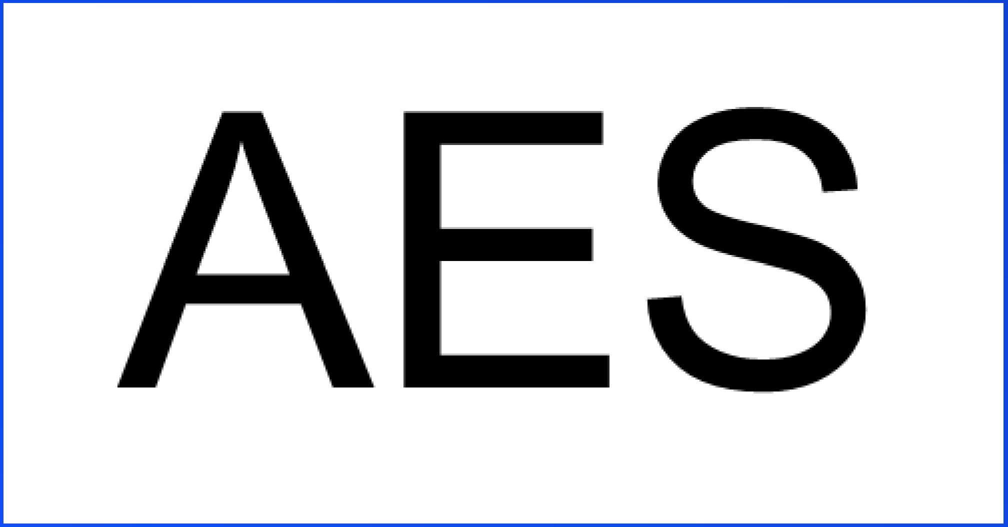
Some Metrics… (ctd)

There were so many other ways to render DKIM ineffective that no-one bothered attacking the crypto

- Anyone with a bit of technical knowledge could have broken the crypto
- No-one did because it was so easy to bypass that it wasn't worth attacking
  - "Crypto is bypassed, …"

Strong crypto will Save Us!

AES-256, because we want keys that go to 11



Original image, unencrypted

Strong crypto will Save Us! (ctd)

AES-256, because we want keys that go to 11

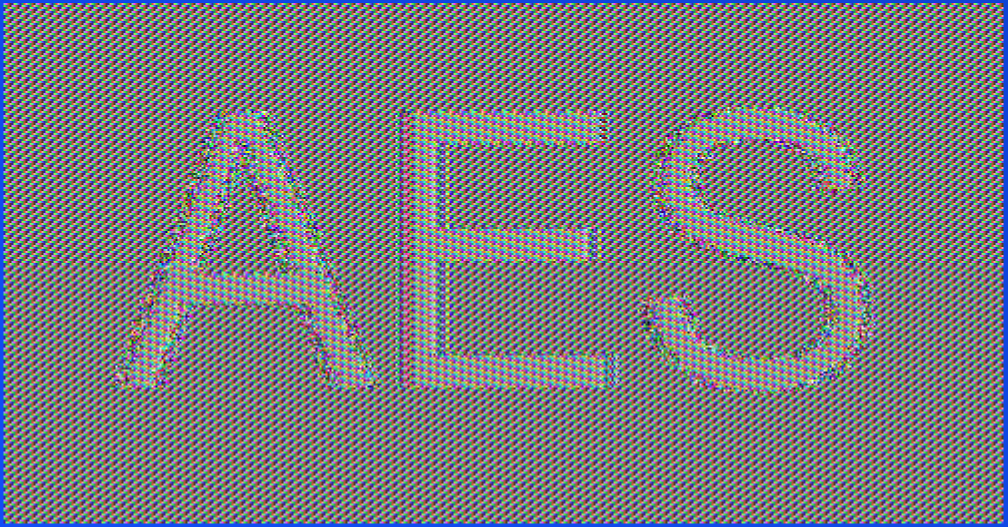


Image encrypted with AES-256, ECB mode

HSMs will Save Us!

Hardware Security Module

- All crypto and keys are locked inside the HSM

Banks use these in large quantities for ATMs and PIN processing

HSMs will Save Us! (ctd)

HSM used for PIN processing

- Encrypt the customer's primary account number (PAN) under the PIN derivation key (PDK) to get the PIN
- Result is a set of values in the range 0x0 – 0xF
- Use a decimalisation table to convert to PIN digits in 0…9 range

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	6

- $\text{encrypt}_{\text{PDK}}$(PAN) = 2A3F…
- Decimalise 2A3F $\rightarrow$ 2036

HSMs will Save Us! (ctd)

Customer-defined PINs are handled by adding an offset to the PIN

- Not security-critical, since it’s useless without the PIN

PIN verification

- Take an encrypted PIN block from the ATM
- Feed it to the HSM in the bank alongside the decimalisation table
- HSM verifies the PIN and returns “failure” or “success”

All inside the HSM

- No keys or plaintext ever leaves the HSM

Secure, right?

HSMs will Save Us! (ctd)

Decimalisation tables are customer-defined

- Use a modified table to guess each PIN digit

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	1	1	2	3	4	5	6	7	8	9	1	1	2	3	4	6

- Enter PIN block
- If the HSM still reports “success” then the PIN contains no zeroes

Repeat for all digits

- Now you know the digits in the PIN, but not their location

HSMs will Save Us! (ctd)

To find the digit locations, adjust the PIN offset

- Use offset to cancel out the decimalisation-table modification

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	1	1	2	3	4	5	6	7	8	9	1	1	2	3	4	6

  – This table converts 0s to 1s in the PIN

- Taking PIN 2036 (from previous slides), offset 0000

Offset	HSM result	PIN
0001	failure	????
0010	failure	????
0100	success	?0??

HSMs will Save Us! (ctd)

Iterate for each digit in the PIN

- Recovers the PIN without knowing any encryption keys or having access to the HSM’s internals

Crypto Summary

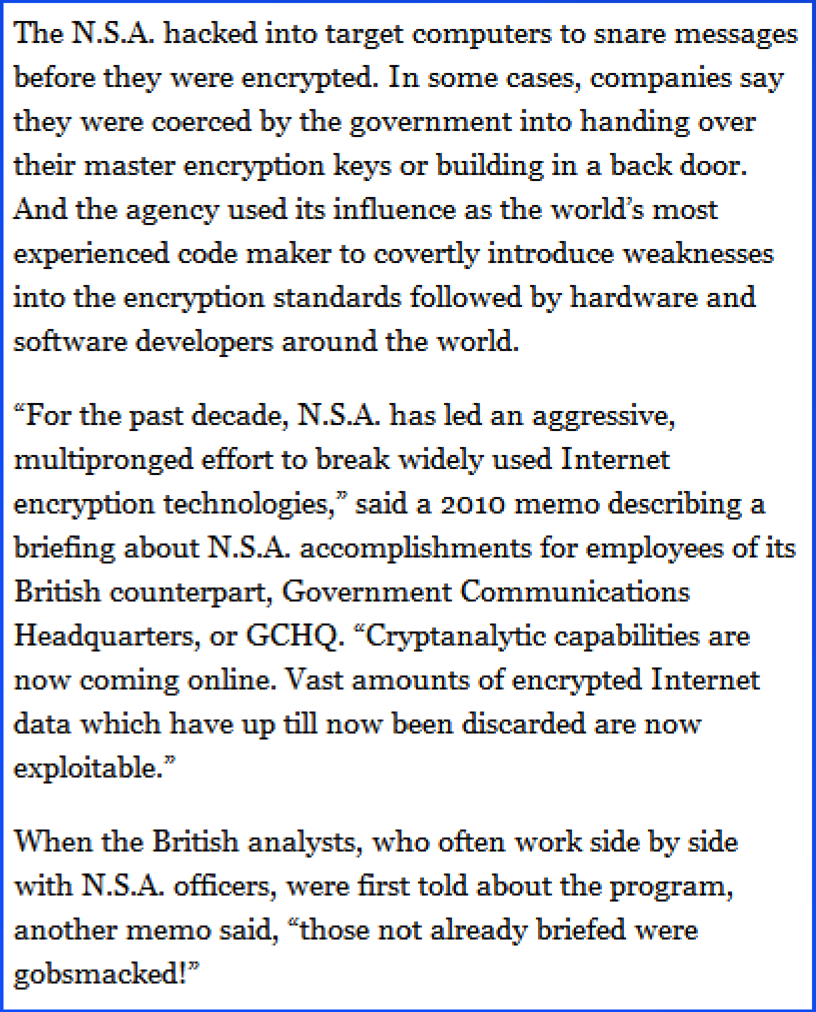
Number of attacks that broke the crypto: 0

Number of attacks that bypassed the crypto: All the rest

- No matter how strong the crypto was, or how large the keys were, the attackers walked around it

Getting Back to BULLRUN…

New York Times:



The N.S.A. hacked into target computers to snare messages before they were encrypted. In some cases, companies say they were coerced by the government into handing over their master encryption keys or building in a back door. And the agency used its influence as the world's most experienced code maker to covertly introduce weaknesses into the encryption standards followed by hardware and software developers around the world.

"For the past decade, N.S.A. has led an aggressive, multipronged effort to break widely used Internet encryption technologies," said a 2010 memo describing a briefing about N.S.A. accomplishments for employees of its British counterpart, Government Communications Headquarters, or GCHQ. "Cryptanalytic capabilities are now coming online. Vast amounts of encrypted Internet data which have up till now been discarded are now exploitable."

When the British analysts, who often work side by side with N.S.A. officers, were first told about the program, another memo said, "those not already briefed were gobsmacked!"

"the NSA hacked into target computers"

"companies were coerced by the government into handing over master encryption keys"

One-week CERT Summary (SB13-273)

“obtain administrative privileges by leveraging read access to the configuration file”, “allows remote authenticated users to bypass an unspecified authentication step”, “allows remote attackers to discover usernames and passwords via an HTTP request”, “allows remote attackers to execute arbitrary commands”, “allows remote attackers to read arbitrary files”, “allows remote attackers to read arbitrary text files”, “allows remote authenticated users to execute arbitrary code”, “allows local users to gain privileges”, “allows remote attackers to obtain sensitive information or modify data”, “allows remote attackers to execute arbitrary SQL commands”, “allows remote attackers to execute arbitrary SQL commands”, “allows local users to gain privileges”, “allows man-in-the-middle attackers to spoof SSL servers”, “allows man-in-the-middle attackers to spoof servers”, “allows man- in-the-middle attackers to obtain sensitive information or modify the data stream”, “allows local users to gain privileges”, “allows remote attackers to enumerate valid usernames”, “allows remote attackers to execute arbitrary commands”, “allows remote attackers to execute arbitrary commands”, “allows local users to execute arbitrary Baseboard Management Controller (BMC) commands”, “allows man-in-the-middle attackers to read or modify an inter-device data stream”, “allows local users to gain privileges”, “allow remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to obtain sensitive query string or cookie information”, “allows remote attackers to hijack the authentication of administrators”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to inject arbitrary web script or HTML”, “allows local users to obtain sensitive information”, “allows remote attackers to conduct cross-site request forgery (CSRF) attacks”, “allows remote attackers to inject arbitrary web script or HTML via an HTML”, “allows remote attackers to execute arbitrary code”, “allows remote attackers to execute arbitrary code”, “allow remote attackers to inject arbitrary web script or HTML”, “allows local users to bypass intended access restrictions”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to obtain sensitive information”, “allows remote attackers to obtain sensitive information”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to read session cookies”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to obtain privileged access”, “allows local users to gain privileges”, “allows remote attackers to execute arbitrary code”, “allows remote attackers to inject arbitrary web script or HTML”, “allows local users to gain privileges”, “allows remote attackers to obtain sensitive information”, “allows remote attackers to inject arbitrary web script or HTML”, “allows local users to gain privileges”, “allows local users to gain privileges”, “allows remote attackers to obtain sensitive information”, “allow remote attackers to bypass intended access restrictions”, “allows remote authenticated users to bypass intended payment requirements”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote attackers to bypass TLS verification”, “allows remote attackers to inject arbitrary web script or HTML”, “allows remote

National Security Letters

The legalised form of rubber-hose cryptanalysis

- Requirement to hand over data, or else
- Built-in gag order to prevent you talking about it
  - Details of both vary depending on court challenges to their constitutionality

National Security Letters (ctd)

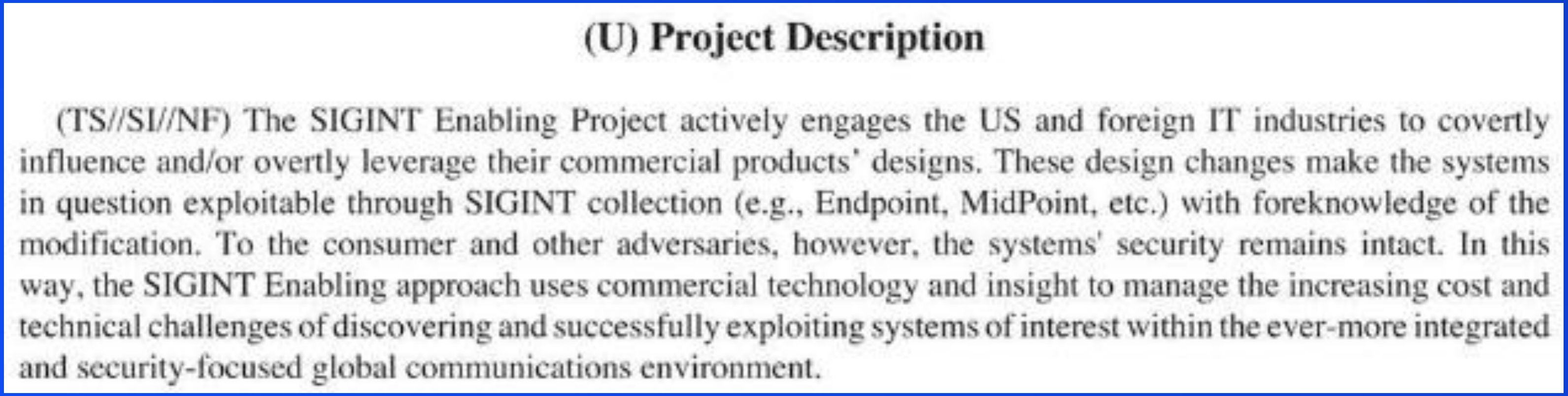
Bypass any crypto at the service provider by requiring them to hand over plaintext

- FBI over-used them while under-reporting their use to Congress

Several providers (LavaBit, Silent Mail, CryptoSeal, CertiVox) have shut down in the face of NSLs

- Larger, more commercially-oriented providers complied with them

BULLRUN Again…



(U) Project Description

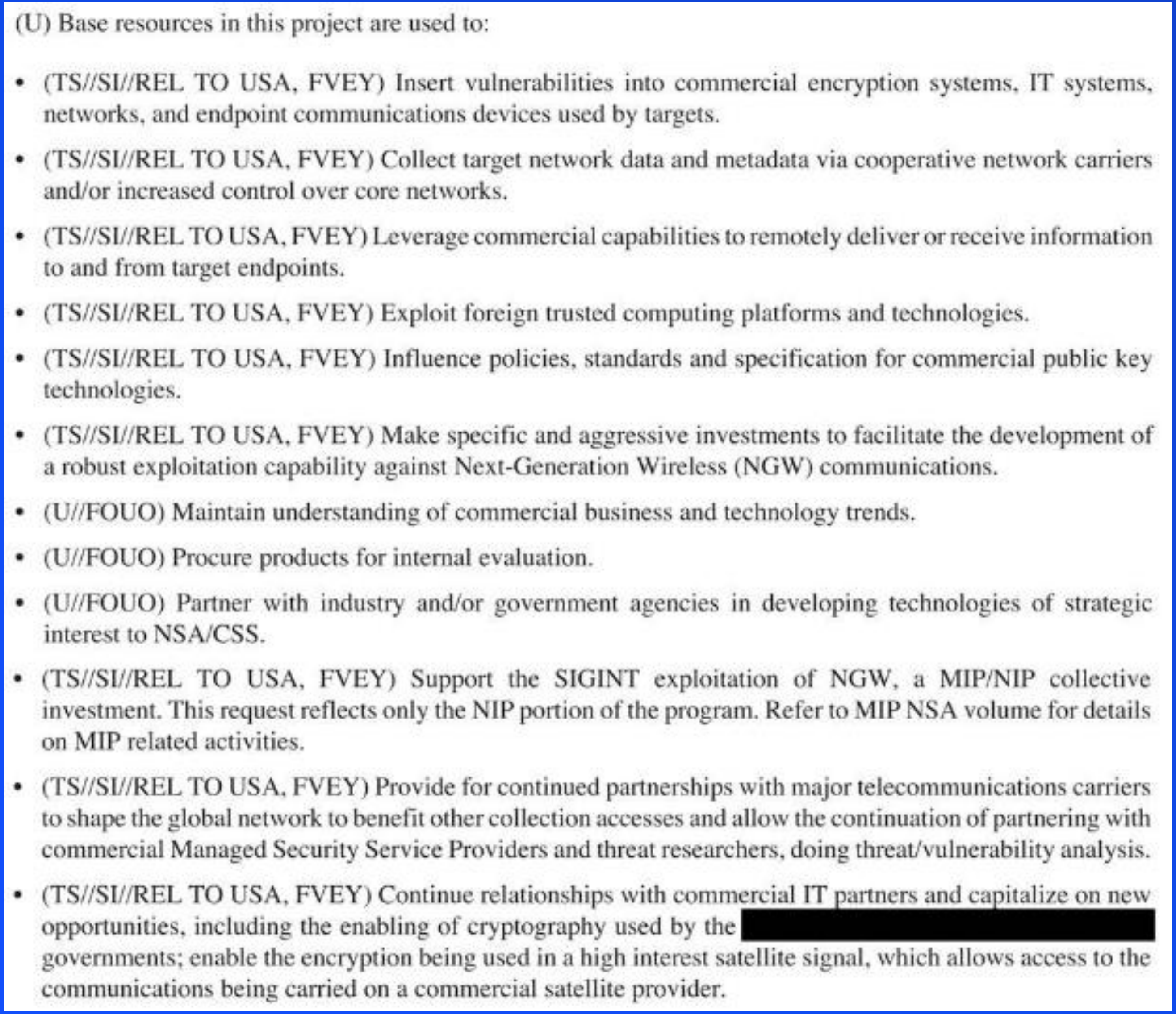
(TS//SI//NF) The SIGINT Enabling Project actively engages the US and foreign IT industries to covertly influence and/or overtly leverage their commercial products' designs. These design changes make the systems in question exploitable through SIGINT collection (e.g., Endpoint, MidPoint, etc.) with foreknowledge of the modification. To the consumer and other adversaries, however, the systems' security remains intact. In this way, the SIGINT Enabling approach uses commercial technology and insight to manage the increasing cost and technical challenges of discovering and successfully exploiting systems of interest within the ever-more integrated and security-focused global communications environment.

“covertly influence and/or overtly leverage commercial products’ designs”

“design changes make the systems in question exploitable”

“to the consumer, however, the systems’ security remains intact”

BULLRUN Again… (ctd)



(U) Base resources in this project are used to:

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.
- (TS//SI//REL TO USA, FVEY) Make specific and aggressive investments to facilitate the development of a robust exploitation capability against Next-Generation Wireless (NGW) communications.
- (U//FOUO) Maintain understanding of commercial business and technology trends.
- (U//FOUO) Procure products for internal evaluation.
- (U//FOUO) Partner with industry and/or government agencies in developing technologies of strategic interest to NSA/CSS.
- (TS//SI//REL TO USA, FVEY) Support the SIGINT exploitation of NGW, a MIP/NIP collective investment. This request reflects only the NIP portion of the program. Refer to MIP NSA volume for details on MIP related activities.
- (TS//SI//REL TO USA, FVEY) Provide for continued partnerships with major telecommunications carriers to shape the global network to benefit other collection accesses and allow the continuation of partnering with commercial Managed Security Service Providers and threat researchers, doing threat/vulnerability analysis.
- (TS//SI//REL TO USA, FVEY) Continue relationships with commercial IT partners and capitalize on new opportunities, including the enabling of cryptography used by the [redacted] governments; enable the encryption being used in a high interest satellite signal, which allows access to the communications being carried on a commercial satellite provider.

Dual_EC_DRBG

In 1985, ANSI X9.17 specified a pseudorandom number generator (PRNG) for banking use

```
temp = encrypt( seed );
out = encrypt( temp ^ $V_n$ );
$V_{n+1}$ = encrypt( out ^ temp );
```

Based on triple DES, the state of the art at the time

- Security relies on the strength of 3DES secret keys

Dual_EC_DRBG (ctd)

In 1998, NIST adopted it verbatim in X9.31, adding the option to use AES

Over a period of several years subsequently, many people at NIST hacked around on a bunch of PRNGs

- Design-by-committee, but in series rather than parallel

Finally published in 2012 as NIST SP 800-90A

Dual_EC_DRBG (ctd)

Some SP 800-90 generators are straightforward and sensible

- X9.17/X9.31 updated to use HMAC
- Half a page in X9.17

Some are not

- Hash_DRBG
- Five pages in SP 800-90

Dual_EC_DRBG (ctd)

Others are just stupid

- Dual_EC_DRBG
- Sixteen pages in SP 800-90
  - Pages and pages of maths
  - Where’s the RNG?
- Complex, awkward, incredibly slow, …

NSA also pushed hard to get it into other standards

- ANSI X9.82
- ISO 18031

These are even worse than SP 800-90

- No way to generate your own parameters

Dual_EC_DRBG (ctd)

It’s OK, no-one in their right mind would implement this

> I’ve never met anyone who would actually use Dual-EC-DRBG. (Blum-Blum-Shub-fanatics show up all the time, but they are all nutcases)
>
> — Kristian Gjøsteen, Norwegian University of Science and Technology

- (Kristian submitted a comment paper to NIST as far back as 2006 pointing out that the EC DRBG was cryptographically unsound and shouldn’t be used)

Dual_EC_DRBG (ctd)

So we’ve established that no-one would ever take this thing seriously

You were serious about dat?

— “My Cousin Vinnie”, 1992

Dual_EC_DRBG (ctd)

Well, except for a pile of US companies, including

- Blackberry
- Certicom (holders of ECC patents)
- Cisco
- GE Healthcare
- Juniper
- Lancope (who *only* provide EC_DRBG)
- McAfee
- Microsoft
- Mocana
- Openpeak

continues

Dual_EC_DRBG (ctd)

continued

- OpenSSL (umbrella use by numerous organisations)
- RSA
- Safenet
- SafeLogic
- Samsung (must have had USG customers)
- Symantec
- Thales (see Samsung entry)

RSA made it the default in their crypto library

Dual_EC_DRBG (ctd)

OpenSSL didn’t actually use it, though

- Implementation contained “a fatal bug in the Dual EC DRBG implementation”

  This bug is fatal in the sense that it prevents all use of the Dual EC DRBG algorithm […] we do not plan to correct the bug. A FIPS 140-2 validated module cannot be changed without considerable expense and effort

  — “Flaw in Dual EC DRBG (no, not that one)”, Steve Marquess

Presumably no-one had ever used this generator in OpenSSL, since no-one complained that it didn’t work

- *Presumably...*

Dual_EC_DRBG (ctd)

FIPS 140 doesn’t allow you to fix things

> We did specifically ask if we had any discretion at all in the choice of points and were told that we were required to use the compromised points […] if you want to be FIPS 140-2 compliant you MUST use the compromised points
>
> — “Flaw in Dual EC DRBG (no, not that one)”, Steve Marquess

But wouldn’t the FIPS validation have caught the fact that the OpenSSL implementation didn’t work?

> Not only the original validation but many subsequent validations have successfully passed the algorithm tests… several hundred times now. That’s a lot of fail […] the FIPS 140-2 validation testing isn’t very useful for catching real-world problems
>
> — “Flaw in Dual EC DRBG (no, not that one)”, Steve Marquess

Dual_EC_DRBG (ctd)

So what's the problem (apart from it being a stupid design)?

- How long do you have?
- Read "The Many Flaws of Dual_EC_DRBG", `http://blog.cryptographyengineering.com/2013/09/the-many-flaws-of-dualecdrbg.html`
- (You are not expected to understand this)

Dual_EC_DRBG (ctd)

Short summary of just one issue

- Public value sent at start of SSL/TLS handshake, Client Random, is 32 bytes (256 bits)
  - Used to randomise each new exchange
- If generated with Dual_EC_DRBG you can predict the SSL/TLS premaster secret
- All crypto keys in SSL/TLS are derived from this value

Dual_EC_DRBG (ctd)

NSA attempted to make this attack even easier

The United States Department of Defense has requested a TLS mode which allows the use of longer public randomness values

- — `draft-rescorla-tls-extended-random-00`
- (Eric Rescorla is co-chair of the TLS working group, draft co-authored by Margaret Salter of the NSA)

- Leaks even more information needed to recover the generator's internal state

Dual_EC_DRBG (ctd)

WTF RSA?

- Specified in a NIST standard
- Lots of government customers
- Implemented several of the generators in the standard
  - Including the dumb ones
- Speculation: “It would really help this large government contract if you made EC_DRBG he default. It’s OK, it’s a NIST-approved generator like all the others”

Dual_EC_DRBG (ctd)

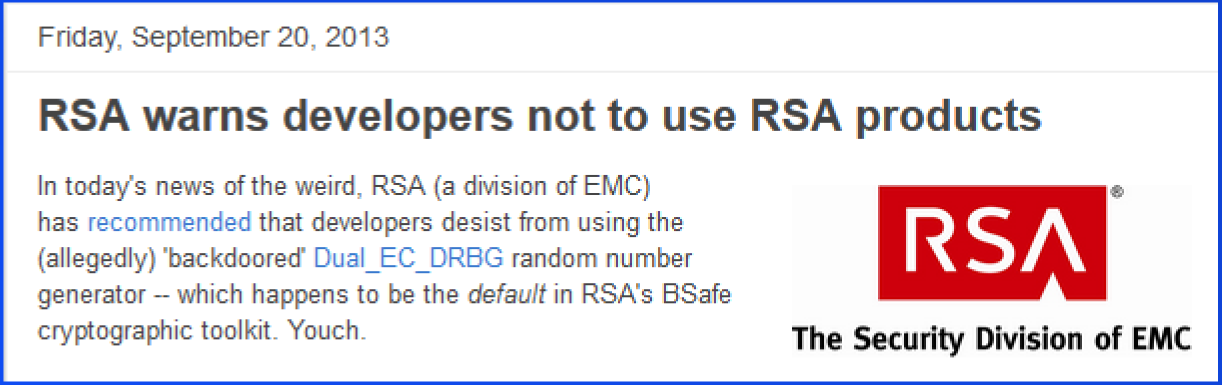
It was more sinister than that though

> RSA received $10 million in a deal that set the NSA formula as the default method for number generation in the BSafe software […] it represented more than a third of the revenue that the relevant division at RSA had taken in during the entire previous year
>
> — Reuters, "Secret contract tied NSA and security industry pioneer"

NSA then used this to force its adoption as a standard

> RSA adopted the algorithm even before NIST approved it. The NSA then cited the early use of Dual Elliptic Curve inside the government to argue successfully for NIST approval
>
> — Reuters, "Secret contract tied NSA and security industry pioneer"

Dual_EC_DRBG (ctd)



Friday, September 20, 2013

RSA warns developers not to use RSA products

In today's news of the weird, RSA (a division of EMC) has recommended that developers desist from using the (allegedly) 'backdoored' Dual_EC_DRBG random number generator -- which happens to be the *default* in RSA's BSafe cryptographic toolkit. Youch.

Dual_EC_DRBG (ctd)

Microsoft's reason for adding it parallels the RSA one (without the bribe):

> Microsoft decided to include the algorithm in its operating system because a major customer was asking for it
>
> — Kim Zetter, Wired

As does OpenSSL's

> It was requested by a sponsor as one of several deliverables. The reasoning at the time was that we would implement any algorithm based on official published standards
>
> — "Flaw in Dual EC DRBG (no, not that one)", Steve Marquess

Dual_EC_DRBG (ctd)

It’s OK though, apart from RSA (and Lancope) no-one made it the default

- It has to be explicitly configured to be the default

Surely no-one would do that

- Except perhaps a large government organisation…

  … the NSA hacked into target computers…

  … to the consumer the systems’ security remains intact…

Just the mere *presence* of such a facility is already a security risk

How to Backdoor Dual_EC_DRBG

Backdoor capability was first pointed out in 2005

> If P and Q are established in a security domain controlled by an administrator, and the entity who generates Q for the domain does so with knowledge of e (or indirectly via knowledge of d), the administrator will have an escrow key for every ECRNG that follows that standard
>
> — "Elliptic curve random number generation",
> Patent Application CA2594670 A1, 21 January 2005

How to Backdoor Dual_EC_DRBG (ctd)

In December 2013, Aris Adamantiadis released OpenSSL-based proof-of-concept code to backdoor the EC_DRBG

> It is quite obvious in light of the recent revelations from Snowden that this weakness was introduced by purpose by the NSA. It is very elegant and leaks its complete internal state in only 32 bytes of output […] It is obviously complete madness to use the reference implementation from NIST
>
> — Aris Adamantiadis, “Dual_EC_DRBG backdoor: a proof of concept”

Used his own EC parameters (not the NIST ones)

- Only the NSA can break the one with the NIST parameters, since it requires knowledge of the secret value *d* used to generate them

NIST ECC Curves

ECC isn't so much an algorithm as a set of toothpicks and a tube of glue

- All the bells, whistles, and gongs you'll ever need

Need to define standardised parameters ("curves") for interoperability

- NIST defined several
- Most common are P256, P384, and P512

NIST ECC Curves (ctd)

Example: P256 curve over a prime field

Prime p = 115792089210356248762697446949407573530086143415290314195533631308867097853951

Parameter a = 115792089210356248762697446949407573530086143415290314195533631308867097853948

Parameter b = 41058363725152142129326129780047268409114441015993725554835256314039467401291

Base point xG = 48439561293906451759052585252797914202762949526041747995844080717082404635286

Base point yG = 36134250956749795798585127919587881956611106672985015071877198253568414405109

Order q of the point G = 115792089210356248762697446949407573529996955224135760342422259061068512044369

- (You are not expected, etc)

NIST ECC Curves (ctd)

How were these generated?

- Deterministically (i.e. verifiably), from a public seed value

What’s the seed value?

- C49D3608 86E70493 6A6678E1 139D26B7 819F7E90

Where did that come from?

- Jerry Solinas at the NSA
- (Jerry is a known ECC mathematician at the NSA)

NIST ECC Curves (ctd)

So how would you use this to backdoor the NIST curves?

- Suppose the NSA knew of (say) a 2^{64} attack that breaks one 256-bit curve in a billion
- The NSA can recognise from the group order whether an attack on the curve will be successful (reasonable assumption)

This isn’t as unlikely as it seems

- Whole classes of elliptic curves are vulnerable to various attacks that make them (relatively) easy to break
- Generating curve parameters is a lengthy, involved process to find one that isn’t vulnerable to the catalogue of known attacks

NIST ECC Curves (ctd)

NSA generates billions of seeds, from which they generate curves until they find one that’s vulnerable to this attack

- Get it adopted as a NIST standard…
- … which is a the de facto standard used by US software vendors …
- … which is the de facto global standard
  - (Speculation courtesy Dan Bernstein)

The curve is “verifiable” in the sense that it was verifiably generated from the seed

- At that point, things stop

Scenario fits the NIST curves

NIST ECC Curves (ctd)

European Brainpool curve designers recognised this in 2005

- The choice of the seeds from which the curve parameters have been derived is not motivated leaving an essential part of the security analysis open.
- No proofs are provided that the proposed curves do not belong to those classes of curves for which more efficient cryptanalytic attacks are possible.

— "ECC Brainpool Standard Curves and Curve Generation"

Brainpool curves compute their seeds from π

- Newer designs like Dan Bernstein's Curve25519 have even more defences built in

NIST ECC Curves (ctd)

In October 2013, RFC 7027 on using the Brainpool curves in TLS was published

- Announced on the TLS mailing list on 15 October 2013

Support added in OpenSSL, cryptlib, PolarSSL on the same day

- Other implementations added support within days

The TLS working group has never moved so quickly on an issue before…

IPsec

It can’t have got that bad by accident

> IPsec was a great disappointment to us […] virtually nobody is satisfied with the process or the result […] the documentation is very hard to understand […] the ISAKMP specifications [the NSA’s main overt contribution to IPsec] contain numerous errors, essential explanations are missing, and the document contradicts itself in various places […] none of the IPsec documentation provides any rationale for any of the choices that were made […] the reviewer is left to guess […]
>
> —“A Cryptographic Evaluation of IPsec”,
> Niels Ferguson and Bruce Schneier,
> from the first 5 pages of 28

You mean they did this *on purpose*?

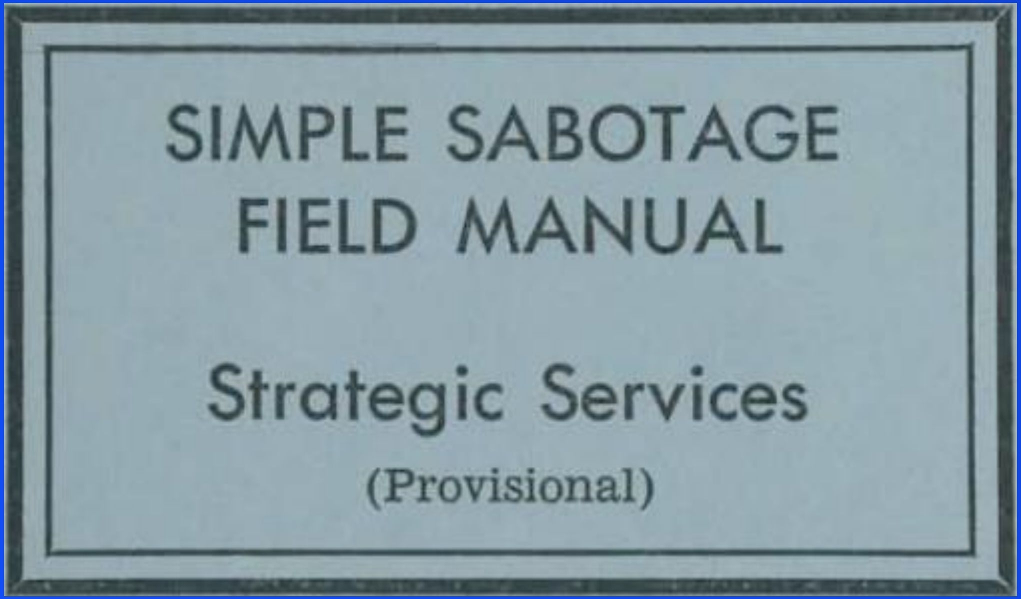
IPsec (ctd)

Hello? I’ve just committed IPsec and I did it on purpose!
— “Last Action Hero”, 1993

Apparently so…

IPsec (ctd)

There's a long history behind this sort of thing

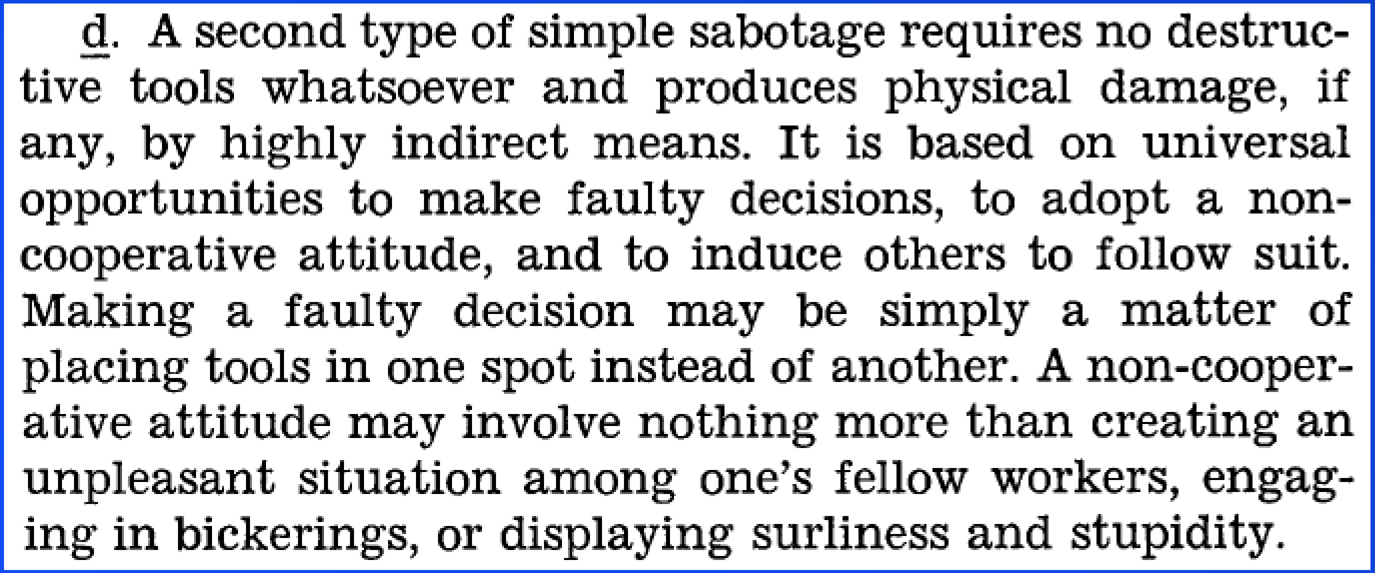


SIMPLE SABOTAGE
FIELD MANUAL

Strategic Services
(Provisional)

OSS field manual, 1945

IPsec (ctd)



d. A second type of simple sabotage requires no destructive tools whatsoever and produces physical damage, if any, by highly indirect means. It is based on universal opportunities to make faulty decisions, to adopt a non-cooperative attitude, and to induce others to follow suit. Making a faulty decision may be simply a matter of placing tools in one spot instead of another. A non-cooperative attitude may involve nothing more than creating an unpleasant situation among one's fellow workers, engaging in bickerings, or displaying surliness and stupidity.

IPsec (ctd)

(a) Organizations and Conferences

(1) Insist on doing everything through "channels." Never permit short-cuts to be taken in order to, expedite decisions.

(2) Make "speeches." Talk as frequently as possible and at great length. Illustrate your "points" by long anecdotes and accounts of personal experiences. Never hesitate to make a few appropriate "patriotic" comments.

(3) When possible, refer all matters to committees, for "further study and consideration." Attempt to make the committees as large as possible - never less than five.

(4) Bring up irrelevant issues as frequently as possible.

(5) Haggle over precise wordings of communications, minutes, resolutions.

(6) Refer back to matters decided upon at the last meeting and attempt to reopen the question of the advisability of that decision.

(7) Advocate "caution." Be "reasonable" and urge your fellow-conferees to be "reasonable" and avoid haste which might result in embarrassments or difficulties later on.

(8) Be worried about the propriety of any decision -raise the question of whether such action as is contemplated lies within the jurisdiction of the group or whether it might conflict with the policy of some higher echelon.

IPsec (ctd)

(b) Managers and Supervisors

(1) Demand written orders.

(2) "Misunderstand" orders. Ask endless questions or engage in long correspondence about such orders. Quibble over them when you can.

(3) Do everything possible to delay the delivery of orders. Even though parts of an order may be ready beforehand, don't deliver it until it is completely ready.

(4) Don't order new working materials until your current stocks have been virtually exhausted, so that the slightest delay in filling your order will mean a shutdown.

(5) Order high-quality materials which are hard to get. If you don't get them argue about it. Warn that inferior materials will mean inferior work.

(6) In making work assignments, always sign out the unimportant jobs first. See that the important jobs are assigned to inefficient workers of poor machines.

(7) Insist on perfect work in relatively unimportant products; send back for refinishing those which have the least flaw. Approve other defective parts whose flaws are not visible to the naked eye.

(8) Make mistakes in routing so that parts and materials will be sent to the wrong place in the plant.

(9) When training new workers, give incomplete or misleading instructions.

(10) To lower morale and with it, production, be pleasant to inefficient workers; give them undeserved promotions. Discriminate against efficient workers; complain unjustly about their work.

(11) Hold conferences when there is more critical work to be done.

(12) Multiply paper work in plausible ways. Start duplicate files.

(13) Multiply the procedures and clearances involved in issuing instructions, pay checks, and so on. See that three people have to approve everything where one would do.

IPsec (ctd)

Hey, I resemble that remark!

- This process may be hard to distinguish from SOP for many organisations

(For people who want this list for use at work:
`http://svn.cacert.org/CAcert/CAcert_Inc/Board/oss/OSS_Simple_Sabotage_Manual.pdf`)

IPsec (ctd)

So was IPsec deliberately sabotaged?

- Probably not

Never attribute to malice what is adequately explained by ~~stupidity~~ a committee

Lesson 1: Cryptographic protocols should not be developed by a committee

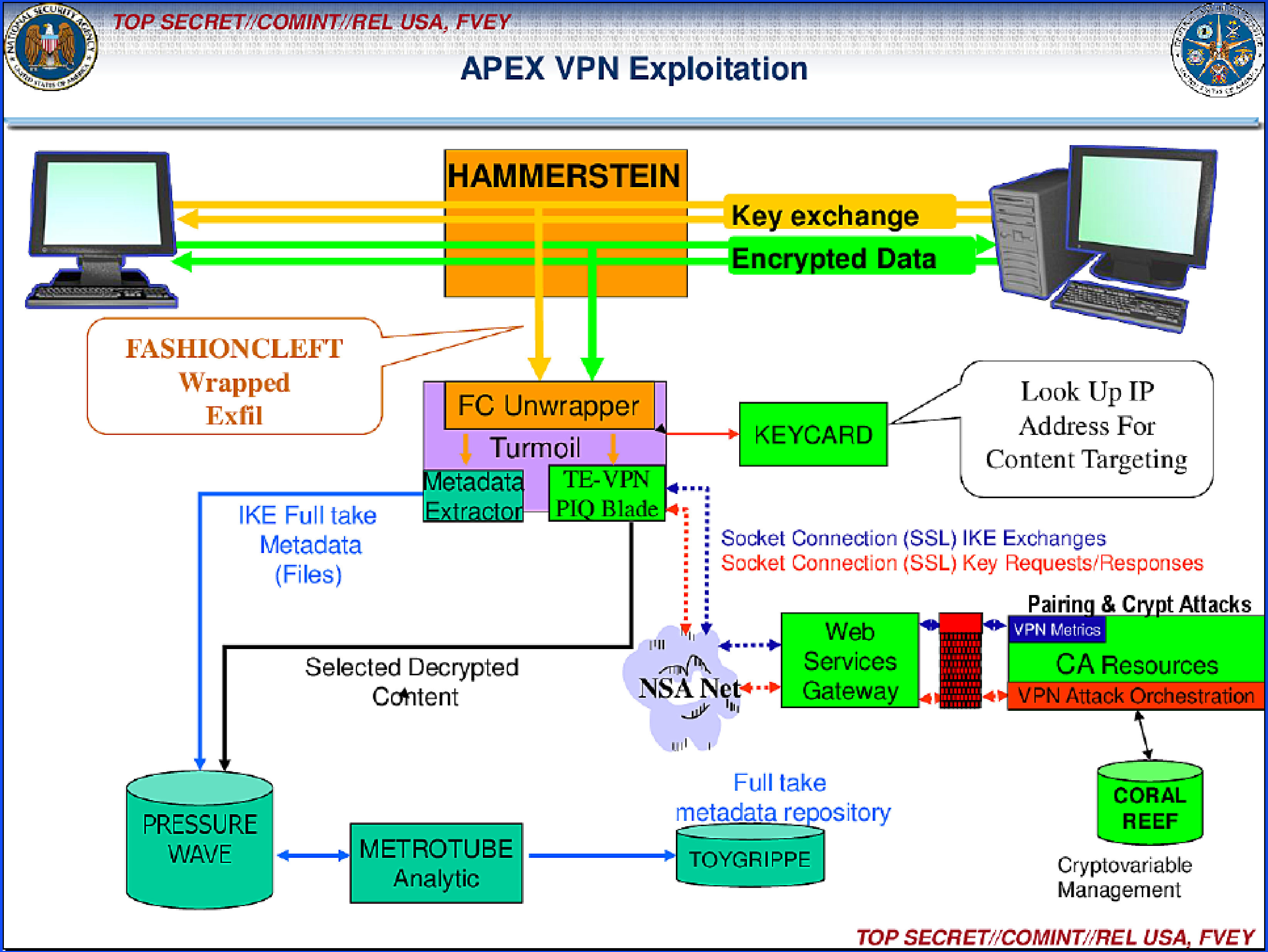
— "A Cryptographic Evaluation of IPsec",
Niels Ferguson and Bruce Schneier

BULLRUN Again…

In any case IPsec doesn’t matter much…

- The NSA have tools for subverting it

BULLRUN Again… (ctd)


TOP SECRET//COMINT//REL USA, FVEY
APEX VPN Exploitation
HAMMERSTEIN
Key exchange
Encrypted Data
FASHIONCLEFT Wrapped Exfil
FC Unwrapper
Turmoil
Metadata Extractor
TE-VPN PIQ Blade
KEYCARD
Look Up IP Address For Content Targeting
IKE Full take Metadata (Files)
Socket Connection (SSL) IKE Exchanges
Socket Connection (SSL) Key Requests/Responses
Pairing & Crypt Attacks
VPN Metrics
CA Resources
VPN Attack Orchestration
Web Services Gateway
NSA Net
Selected Decrypted Content
PRESSURE WAVE
METROTUBE Analytic
Full take metadata repository
TOYGRIPPE
CORAL REEF
Cryptovariable Management
TOP SECRET//COMINT//REL USA, FVEY

BULLRUN Again… (ctd)

As well as the routers that run it…

- When you own the router that does the crypto, IPsec becomes irrelevant

NSA owns

- Cisco
  - BANANAGLEE, JETPLOW
- Juniper
  - BANANAGLEE, FEEDTROUGH, GOURMETTROUGH, SCHOOLMONTANA, SIERRAMONTANA, SOUFFLETROUGH, VALIDATOR
- Huawei
  - HAMMERMILL, HALLUXWATER, HEADWATER

BULLRUN Again… (ctd)

Speaking of routers and security risks…

Q: Does Huawei represent an unambiguous national security threat to the US and Australia?
A: Yes, I believe it does

— NSA Director Michael Hayden, interviewed in the Australian Financial Review

Chinese telecom provider Huawei represents an unambiguous national security threat to the United States and Australia

— "Huawei Is a Security Threat and There's Proof, Says Hayden", eWeek

We'd better go with (expensive) US networking equipment, since we can't trust (cheaper) Huawei gear

BULLRUN Redux

So this…

Chinese telecom provider Huawei represents an unambiguous national security threat to the United States and Australia

— "Huawei Is a Security Threat and There's Proof, Says Hayden", eWeek

… is really this:

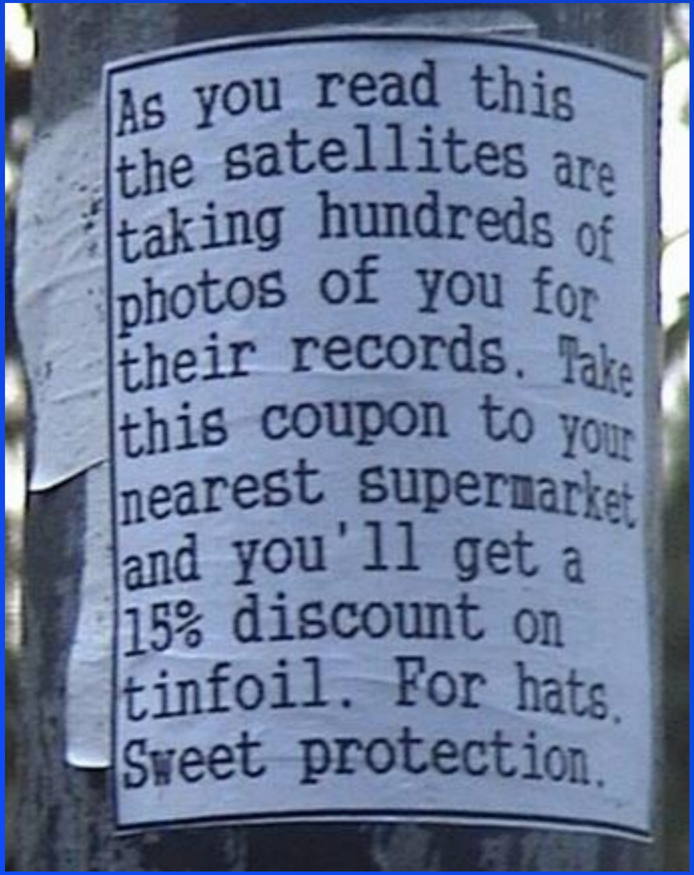
US intelligence agency NSA represents an unambiguous national security threat to the United States and Australia

— "NSA Is a Security Threat and There's Proof, Says Snowden", TBA

NSA-proof Crypto

We don't need any new "NSA-proof protocols"

- Any well-designed, appropriately-deployed protocol is "NSA-proof"


As you read this the satellites are taking hundreds of photos of you for their records. Take this coupon to your nearest supermarket and you'll get a 15% discount on tinfoil. For hats. Sweet protection.

NSA-proof Crypto (ctd)

Any properly-designed and implemented protocol will stop

- The NSA
- The CIA
- The GCSB
- The FSB (née KGB)
- …
- Your mother
- Your cat

NSA-proof Data

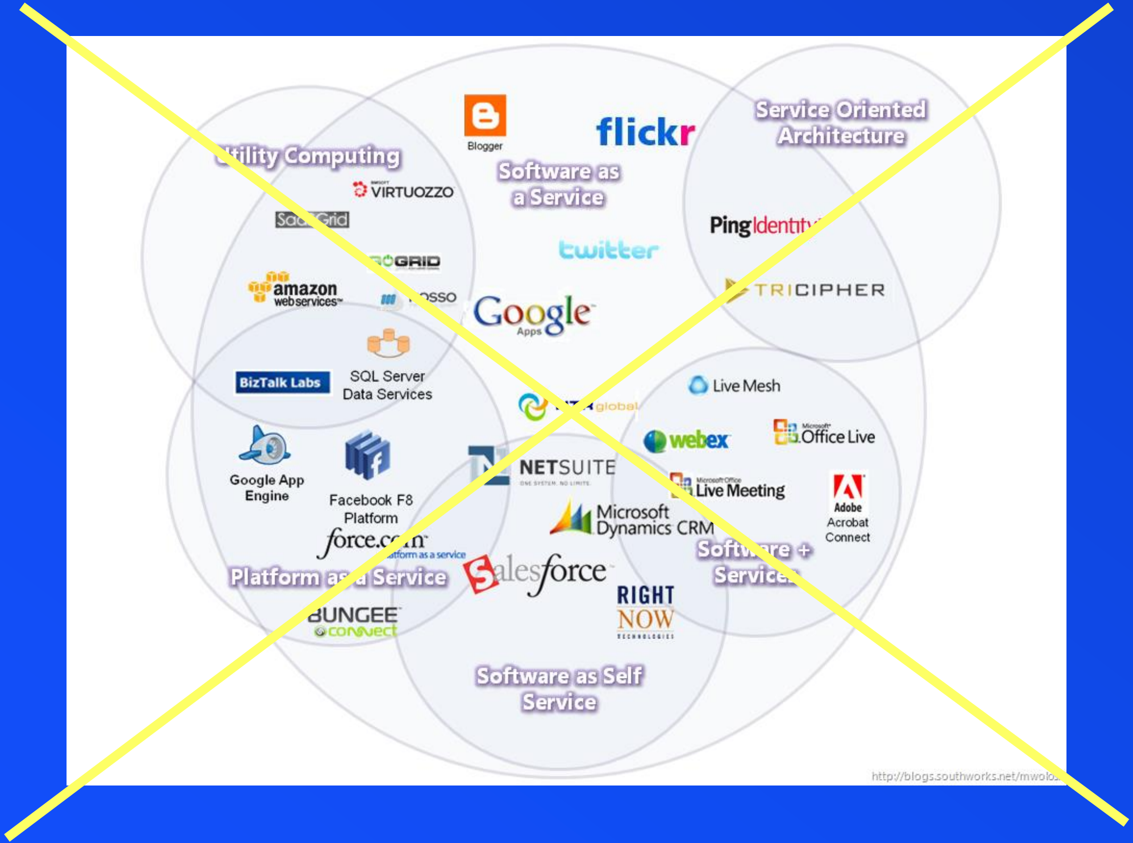
Sometimes we don’t need crypto at all

Utility Computing
Software as a Service
Service Oriented Architecture
Platform as a Service
Software + Services
Software as Self Service
Blogger
flickr
VIRTUOZZO
SaaSGrid
GOGRID
amazon webservices
MOSSO
twitter
Google Apps
PingIdentity
TRICIPHER
BizTalk Labs
SQL Server Data Services
NTR global
Live Mesh
webex
Office Live
Google App Engine
Facebook F8 Platform
NETSUITE
Live Meeting
Microsoft Dynamics CRM
Adobe Acrobat Connect
force.com platform as a service
salesforce
RIGHT NOW
BUNGEE connect
http://blogs.southworks.net/mwoloski

Let’s leverage the synergy of the cloud!

NSA-proof Data (ctd)

On second thoughts…



Let’s not.

NSA-proof Data (ctd)

Leverage the safety of your local server

- Getting data from Gmail via an NSL is much easier than getting it from a PC at 81 Princes St, Putaruru 3411, New Zealand

(Counterpoint: Google is better at running a mail server than most companies are)

NSA-proof Data (ctd)

Goes back to a pre-crypto principle called geographic entitlement

- More modern term: location-limited channel

You have to be at least this close to the data in order to access it

- Works best with short-range links, not long-distance routable protocols

NSA-proof Data (ctd)

Access to data is predicated on physical access to the location

NSA-proof Data (ctd)

In plain English: Don’t put your data where the NSA can get it

There’s already pushback in Europe against exporting data to the US

- (So now only your local spooks can get it)

Conclusion

I love crypto, it tells me what part of the system not to bother attacking

— Drew Gross, forensic scientist

Crypto is not soy sauce for security

— Patrick McKenzie

Crypto is fundamentally unsafe. People hear that crypto is strong and confuse that with safe. Crypto can indeed be very strong but it's extremely unsafe

— Nate Lawson, Root Labs

Encryption is the chicken soup of security, feel free to apply it if it makes you feel better because it's not going to make things any worse, but it may not make things any better either

— Me