Countering Cryptographic Subversion

Post-Snowden Cryptography Workshop
Brussels 8/12/2015

Kenny Paterson

Information Security Group

@kennyog ; www.isg.rhul.ac.uk/~kp

ROYAL HOLLOWAY UNIVERSITY OF LONDON

The post-Snowden adversary

- Since the Snowden revelations beginning in 2013, we've seen the emergence of a new cryptographic adversary.
  - One capable of subversion of cryptographic algorithms, standards, and deployed systems.
  - One engaged in offence in depth.
  - Much more powerful than our cute cartoon pictures tend to suggest.
- What is the nature of the subversion? What can we do about it?

UK world politics sport football opinion culture business lifestyle fashion environment tech travel

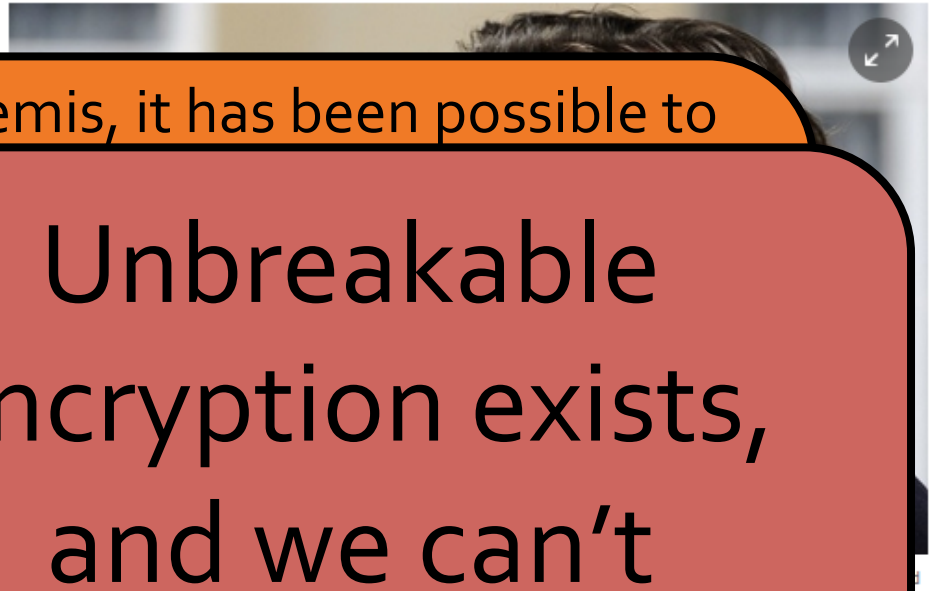
home › tech

Encryption

How has David Cameron caused a storm over encryption?

What did the prime minister say and why has it gone down to badly? Your questions, answered

Alex Hern



"In extremis, it has been possible to

th

Unbreakable encryption exists, and we can't uninvent it.

government would be cracking down o… …nce techniques.

"In extremis, it has been possible to read so… …etter, to listen to someone's call, to listen in on mobile communications," he … "The question remains: are we going to allow a means of communications wh… it simply is not possible to do that? My answer to that question is: no, we must not."

My personal position

If you outlaw strong cryptography, pretty soon the only people using strong cryptography will be outlaws.

Anon

The crypto genie is out of the bottle and he's not going back in again.

"Strong cryptographic algorithms and secure protocol standards are vital tools that contribute to our national security and help address the ubiquitous need for secure, interoperable communications."

NSA, August 2015

The Snowden revelations and cryptography

- The Snowden revelations have not told us that much about the *cryptanalytic* capabilities of NSA/GCHQ.
- "Significant cryptanalytic breakthrough" circa 2008/2009.
  - Speculation: advance in discrete logs? RC4?
- Persistent rumours about real-time breakage of RC4.
- Ability to break certain unnamed VPN products.
- Project BULLRUN: Dual_EC_DBRG.

Related cryptographic issues

- Still widespread deployment of export-grade crypto (exploited in FREAK and LOGJAM attacks).
- NSA's ability to demand or otherwise procure server-side RSA keys.
- Poor key generation practices across the industry – repeated keys, weak keys.
- Fragility of DSA/ECDSA to randomness failures.
- Fragility of AES-GCM.
- Micali-Schnorr: RSA-based PRNG with parameters of dubious provenance (ANSI and ISO standards).
- Unknown provenance of NIST elliptic curves.
- Poor quality of certificate processing code across a wide range of implementations.
- Lack of protection of meta-data in deployed secure communications protocols.

A general point

- Maybe breaking the crypto is not the most effective way to get at data.
- "Just" do CNE instead.
- So what's the point of trying to make our crypto stronger?
- We have many holes to plug, and they are of many different types – OS security, networking, software security, crypto, law and constitutional reform,…
- We eventually want – and need – to plug them all.
  - cf. DNS versus SNI protection debate on TLS mailing list.
- So let's get started!

What can we do?

Illustrative list:

- Definitively break algorithms and protocols suspected to be weak. (Then publicise the work relentlessly.)
- Participate meaningfully in standards development organisations (SDOs) to make better standards.
- Produce free, strong, fast, developer-friendly cryptographic implementations.

Definitive breakage

Examples:

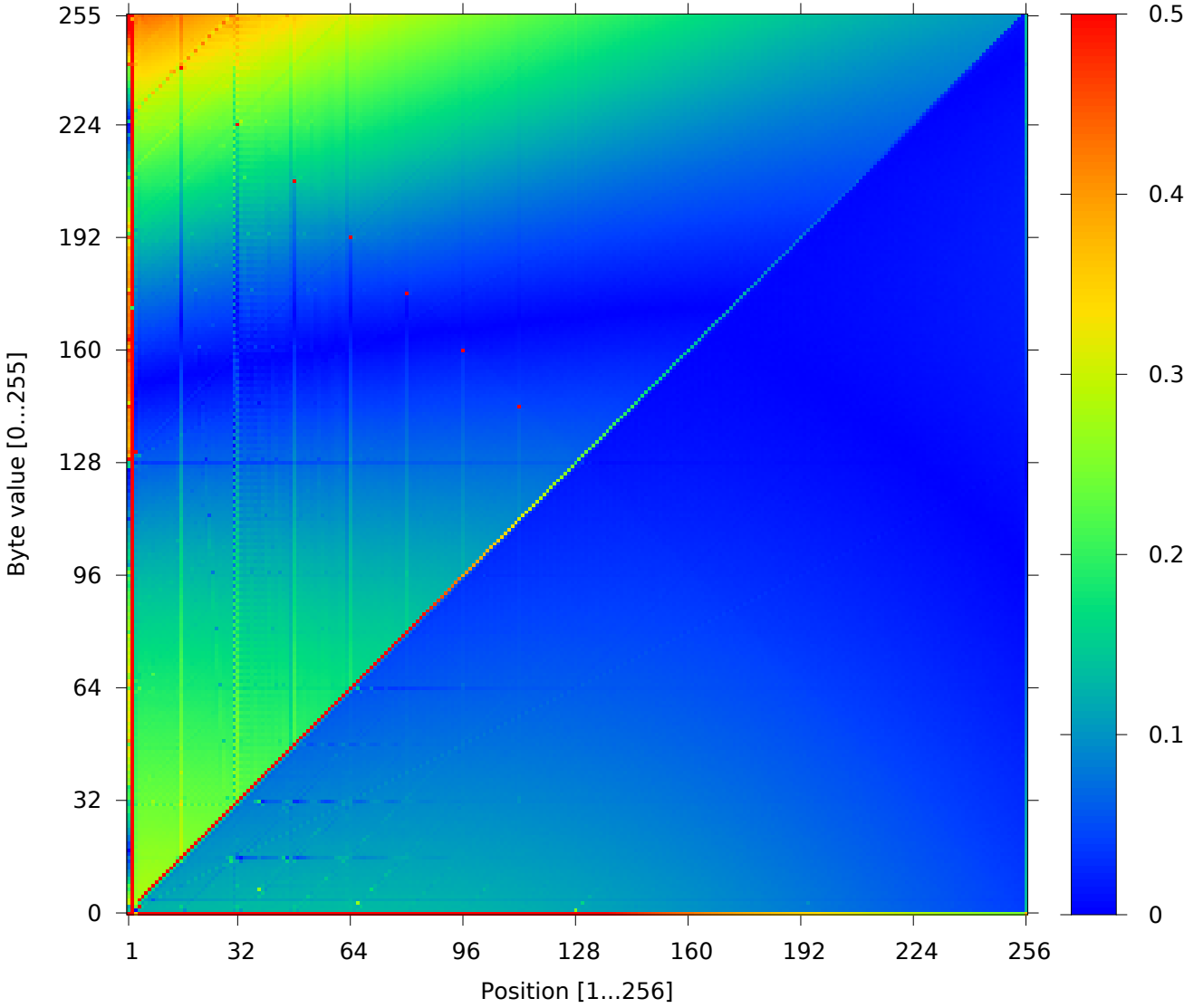
- RC4
- Dual_EC_DBRG
- Export ciphersuites for SSL/TLS

RC4

- In early 2013, roughly 50% of all SSL/TLS traffic was using RC4 for encryption.
- **Breaking news: RC4 is no longer a state-of-the-art stream cipher!**

- Eurocrypt 2012:
  - Dan Bernstein, Tanja Lange and I had a conversation about Lucky 13, an attack against CBC mode in TLS.
  - Dan: what about Rc4 then?
- RWC 2013:
  - Eric Rescorla: CBC bad but we still have RC4 (paraphrase).
  - Dan: what about Rc4 then?

RC4 biases, based on 2^{45} keystreams



Breaking RC4 harder – and harder

[ABPPS13]: use Fluhrer-McGrew biases, 2^{34} encryptions, 2000 hours to recover session cookie.

Jan. 2015: RFC 7465 "Prohibiting RC4 ciphersuites"

> `This document requires that TLS clients and servers never negotiate the use of RC4 cipher suites.`

[GPV15]: refinement of [ABPPS13] attacks focussed on password recovery from early in the keystream: 60% success rate with 2^{26} encryptions, 350 hours.

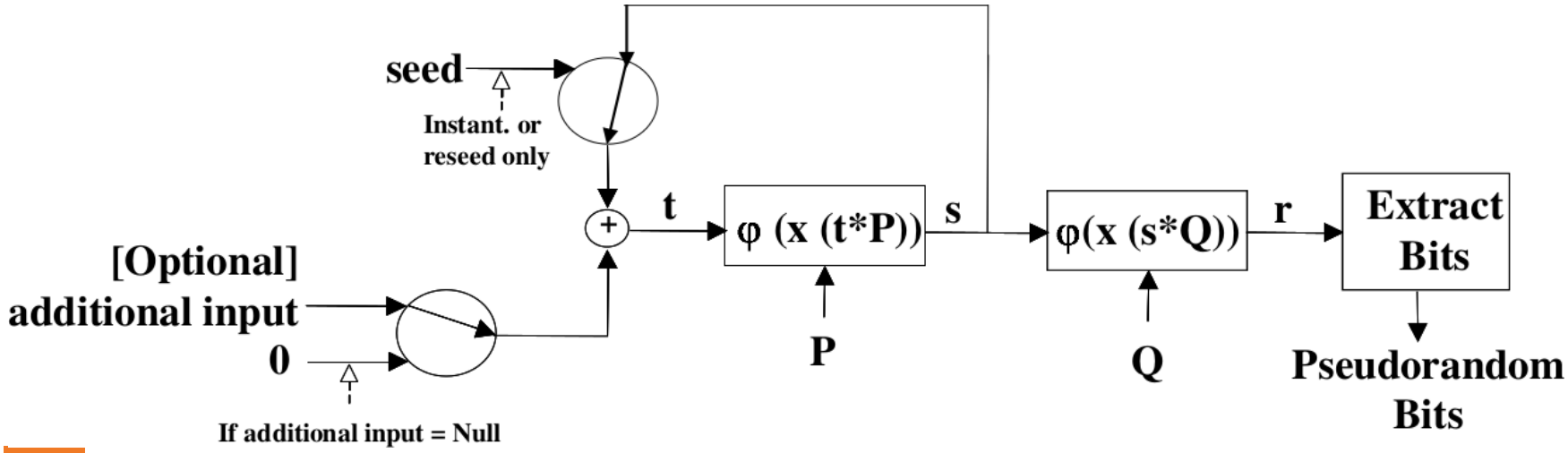
[VP15]: use of Mantin biases to recover cookies: 94% success rate with $2^{30}+2^{27}$ encryptions, 75 hours.

September 1st, 2015: Microsoft, Google, Mozilla all announce that Rc4 will be fully disabled in their browsers in early 2016.

December, 2015: RC4 usage in TLS down to circa 7%.

Dual_EC_DBRG

- Invented by Certicom research staff.
- Very slow, has biased output, clearly *backdoorable* (Shumow-Ferguson, Crypto 2007 rump session).
- In NIST SP 800-90A, along with sensible options.
- No-one would use it, right?



Dual_EC_DBRG

- New York Times reported that NSA had backdoored Dual_EC_DBRG in NIST and ISO standards.
- NSA reported to have paid RSA-DSI to make it the default generator in their BSAFE crypto library.
- Checkoway et al., USENIX Security 2014:
  - With variable computational effort, DualEC is exploitable in the context of the TLS protocol.
  - Leading to complete key recovery attack for TLS sessions.
  - Extensive reverse-engineering and analysis of software that uses the Dual_EC algorithm.
  - Full paper and summary at `dualec.org` and `https://projectbullrun.org/dual-ec/index.html`
- Wider impact: VCAT review, re-evaluation of relationship between NSA and NIST.

Export ciphersuites for SSL/TLS

`EXPORT` ciphersuites:

```
0x000003    TLS_RSA_EXPORT_WITH_RC4_40_MD5
0x000006    TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
0x000008    TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
0x00000B    TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
0x00000E    TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
0x000011    TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
0x000014    TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
```

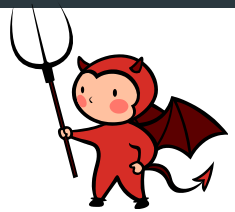
(and more)

- Introduced in the 1990's in the era of export control.
- Maximum 512-bit RSA keys and 512-bit primes for DH/DHE.
- Repurpose `ServerKeyExchange` message to transport "ephemeral" RSA/DH/DHE keys.
- Until recently, still supported by around 25% of servers…

FREAK Attack (Beurdouche et al, IEEE S&P 2015)

ClientHello

TLS_RSA…



ClientHello'

TLS_RSA_EXPORT…

Server accepts TLS_RSA_EXPORT…

ServerHello,
Cert,
ServerKeyExchange

Changed by MITM back to TLS_RSA…

ServerHello',
Cert,
ServerKeyExchange

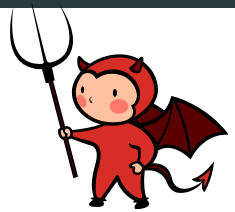
Buggy client processes this and accepts 512-bit RSA key for transport of premastersecret

Contains server's 512-bit RSA public key and RSA signature on nonces and parameters

FREAK Attack (Beurdouche et al, IEEE S&P 2015)

ClientHello

TLS_RSA…



ClientHello'

TLS_RSA_EXPORT…

ServerHello,
Cert,
ServerKeyExchange

ServerHello',
Cert,
ServerKeyExchange

ClientKeyExchange,
CCS,
ClientFinished

Attacker pre-factors 512-bit RSA key, and can now decrypt to get premaster secret.

CCS,
ServerFinished

Attacker succeeds in impersonating server.

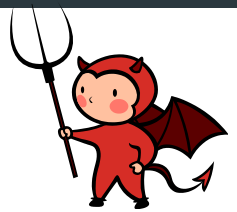
FREAK Attack (Beurdouche et al, IEEE S&P 2015)

- Attack relies on buggy clients accepting ServerKeyExchange containing 512-bit RSA key when no such message was expected.
    - Many clients were vulnerable (https://www.smacktls.com/).
- Export RSA keys are meant to be ephemeral, but it's hard to generate RSA moduli in practice, so they were made long-lived.
- Cost of factoring 512-bit modulus: $50 on Amazon EC2 (Valenta et al, eprint 2015/1000).
- Attack arises because of common code paths in implementations, coupled with state machine failures.
    - Explored in-depth in Berdouche et al paper.

LOGJAM Attack (Adrian et al, ACM-CCS 2015)

ClientHello

TLS_DHE_RSA…



ClientHello'

TLS_DHE_RSA_EXPORT…

ServerHello,
Cert,
ServerKeyExchange

Contains server's 512-bit **DHE parameters** and RSA signature on nonces and parameters

ServerHello',
Cert,
ServerKeyExchange

Attacker uses x and g^y to compute master secret

ClientKeyExchange (g^y), CCS, ClientFinished

Attacker solves DLP for g, g^x to compute server's private value x .

CCS,
ServerFinished

Attacker succeeds in impersonating server.

LOGJAM Attack (Adrian et al, ACM-CCS 2015)

- LOGJAM = Cross-ciphersuite + FREAK.
  - Active attacker changes `TLS_DHE_RSA...` to `TLS_DHE_RSA_EXPORT...`
  - Server responds with weak DH parameters signed under its RSA key.
  - Client accepts these (signature does not include ciphersuite details).
  - Attacker solves 512-bit DLP before client times out.
  - Attacker can then create correct `ServerFinished` message to impersonate server.
- Difficult to perform in practice, but not impossible.
  - Servers use small number of common primes *p.*
  - Precomputation allows each 512-bit DLP to be solved in around 90 seconds.

Characteristics of definitive breakage

- It's hard and messy work.
  - Analysis of specifications, source code, reverse engineering, detective work, implementation/PoC.
  - It's not traditional cryptanalysis, like "Breaking 3 rounds of `blah_blah` with 2^{124} chosen plaintexts"!
- It's under-appreciated by large parts of the crypto research community.
- It is sometimes dismissed as being "incremental" or "not surprising".
- It takes time to have an effect, but the cumulative effects can be significant in improving security of deployed systems.
- Definitive breakage provides irresistible pressure to change systems.

SDO participation

- Participate meaningfully in standards development organisations (SDOs) to make better standards.
- Meaningful = sustained + persuasive + respectful.

- Difficult to impossible for ISO – (see PLAID episode, `eprint.iacr.org/2014/728.pdf`).
- Realistic for NIST (e.g. AES, SHA-3, now new elliptic curves).
- Perfectly do-able for IETF/IRTF/CFRG.

IRTF/CFRG

- IRTF = Internet Research Task Force.
- CFRG = Crypto Forum Research Group.
  - Working on new curves, DH and signature schemes for TLS 1.3, commissioned by the TLS Working Group.
  - Gearing up for work on post-quantum primitives.
- Physical meetings 3 times per year, now exploring co-location with RWC and crypto conferences.
- Main business conducted on mailing list:
  - `https://www.ietf.org/mail-archive/web/cfrg/current/`
  - `cfrg@irtf.org`
  - Anyone can join in; be ready for vigorous discussion.
- We need more help!

Effective cryptographic implementation

- Produce free, strong, fast, developer-friendly cryptographic implementations.
- This is not my particular strength!
- A quick illustration of things we can and should avoid: certificate processing bugs.

Certificate Processing Bugs

Many fatal bugs have been discovered in code for certificate processing.

- Fahl et al. (CCS 2012)
- Georgiev et al. (CCS 2012)
- GnuTLS bug (CVE-2014-0092)
- Apple goto fail (CVE-2014-1266)
  - Affecting Apple iOS 6.x before 6.1.6 and 7.x before 7.0.6, Apple TV 6.x before 6.0.2, and Apple OS X 10.9.x before 10.9.2.
- Frankencerts (IEEE S&P 2014)

Apple `goto fail`

```
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus    err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) !=
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hash                    HashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

Causes all server signature processing on client to be bypassed!

Meaning that MITM attacker can trivially spoof ***any*** TLS server!

Closing remarks

- Theory has a role, but let's not invent new theory for theory's sake, please.
- Recommended reading: Phil Rogaway's recent essay "The Moral Character of Cryptographic Work".

- Keep in mind that what unites in our endeavours is stronger than the things that divide us.