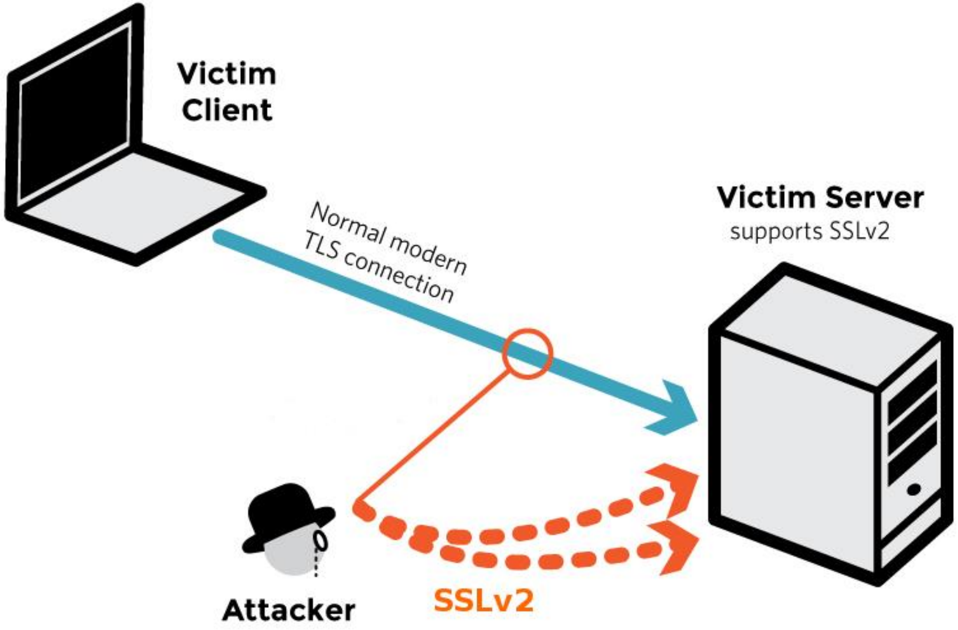
DROWN - Breaking TLS using SSLv2

Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, Yuval Shavitt

A history of obsolete crypto

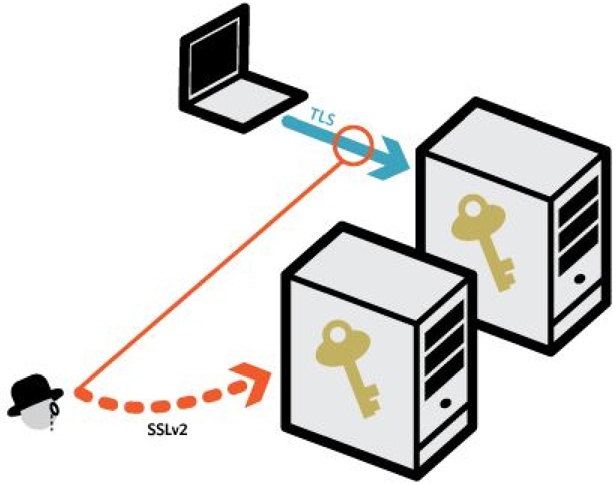
- SSLv2 published in 1995, immediately broken
  - Devastating MitM attacks
  - Common wisdom: SSLv2 is better than plaintext
- Before DROWN: OK to keep SSLv2 enabled, esp. for email.

Our results: SSLv2 breaks TLS

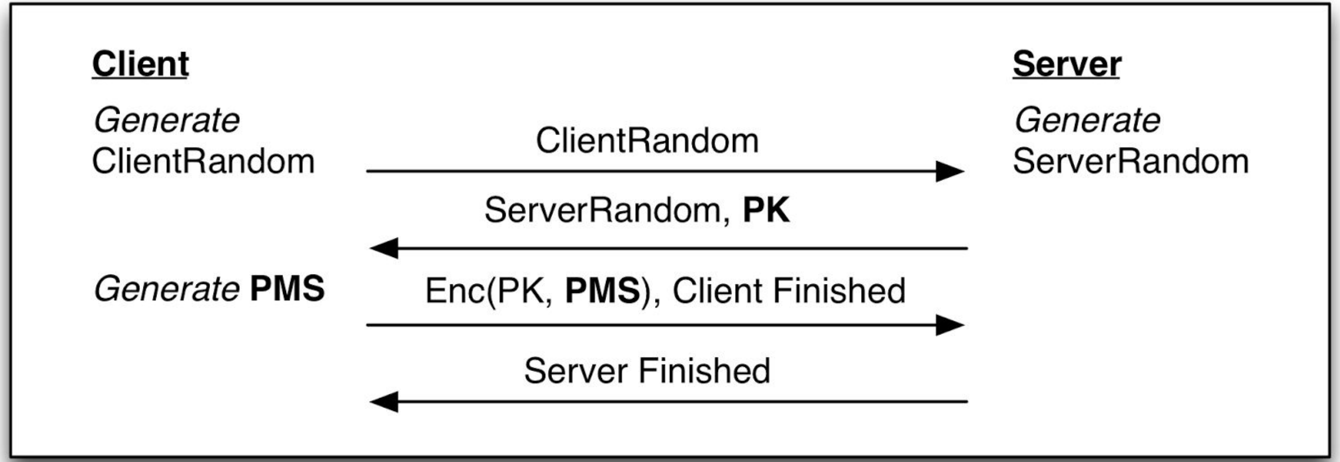


DROWN - Overview

- Attacker decrypts intercepted TLS traffic
- Cross-protocol attack
  - **Attack TLS server using SSLv2 server**
  - Attack HTTPS server using email server - SSLv2 much more prevalent on email ports
- **22% of trusted HTTPS hosts vulnerable** with cross-protocol use

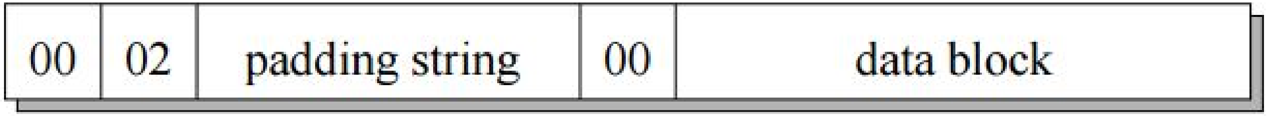


TLS RSA Handshake



PKCS #1 v1.5

- Textbook RSA: k^e mod N
  - Problem: No randomization
- In real-life:
  - PKCS #1 v1.5: pad k to length of N with random padding

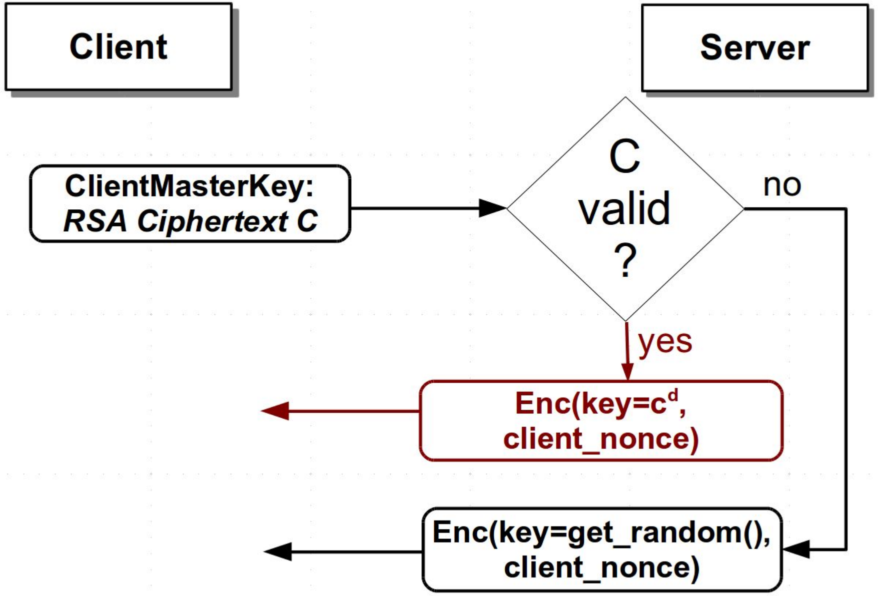


Bleichenbacher's Attack

- If padding is incorrect after decryption, then…
  - ~~Send an error message~~
  - Attacker can deduce if padding was correct.
- Conclusion: **The server has to behave as if the padding was valid!**

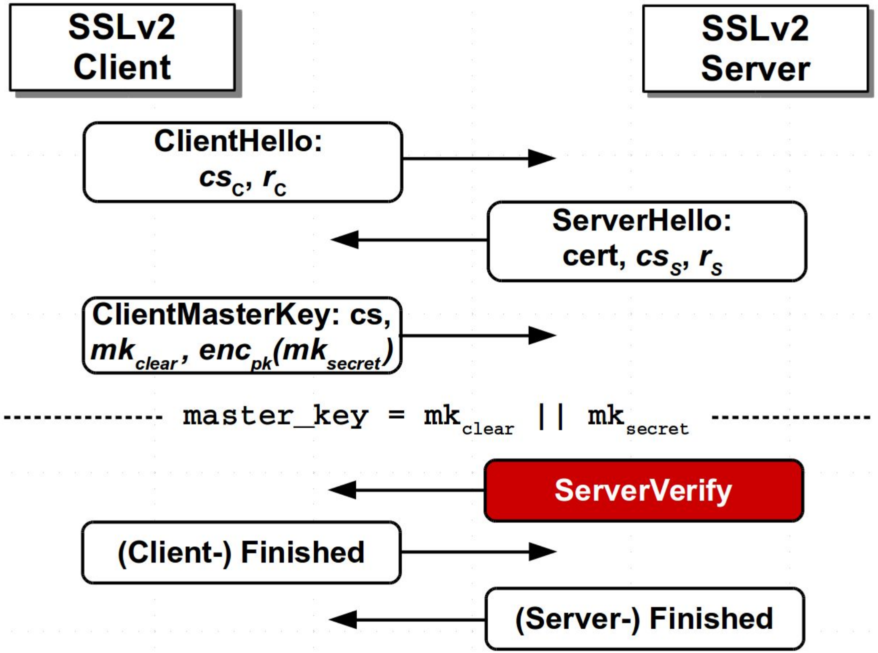
Bleichenbacher's Attack

- If padding is incorrect after decryption, then…
  The server has to behave as if the padding was valid!
- Solution: Server generates a random "replacement" plaintext, continues as usual.



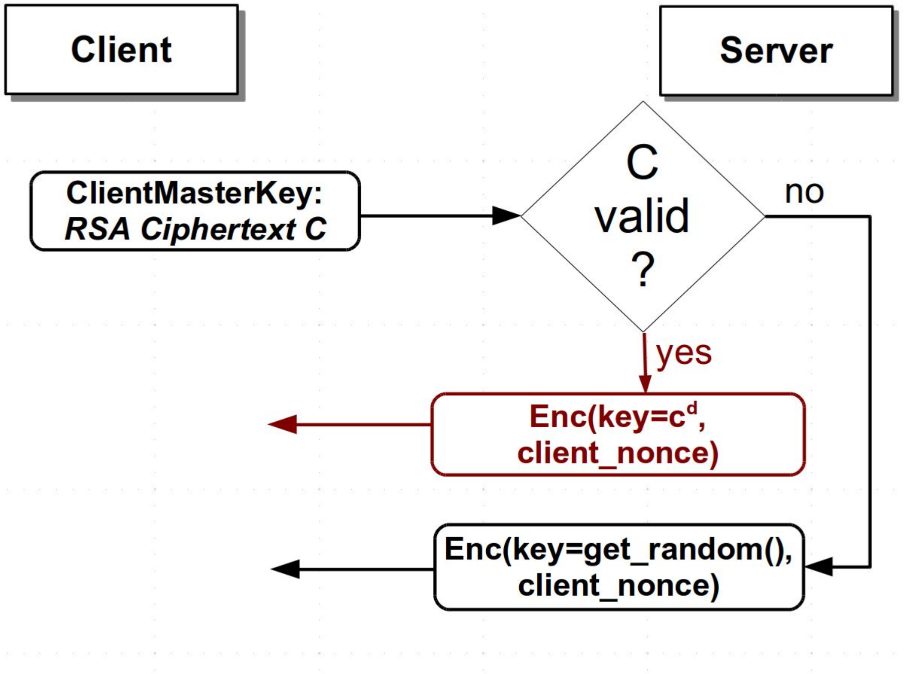
Differences between SSLv2 and TLS

- Server authenticates **first** (sends first message encrypted with symmetric key)
- Short secrets for export grade crypto:
  - SSLv2: 40 bit key.
  - TLS: 48 byte (384 bit) key.



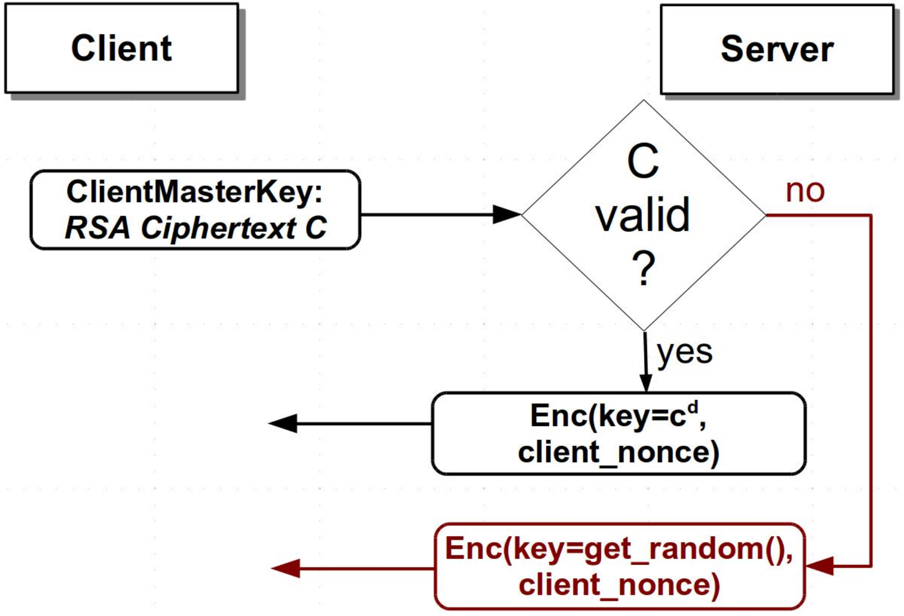
An important observation

- Attacker connects twice with same RSA ciphertext.
- Ciphertext valid:
- 2 server replies encrypted with same key.



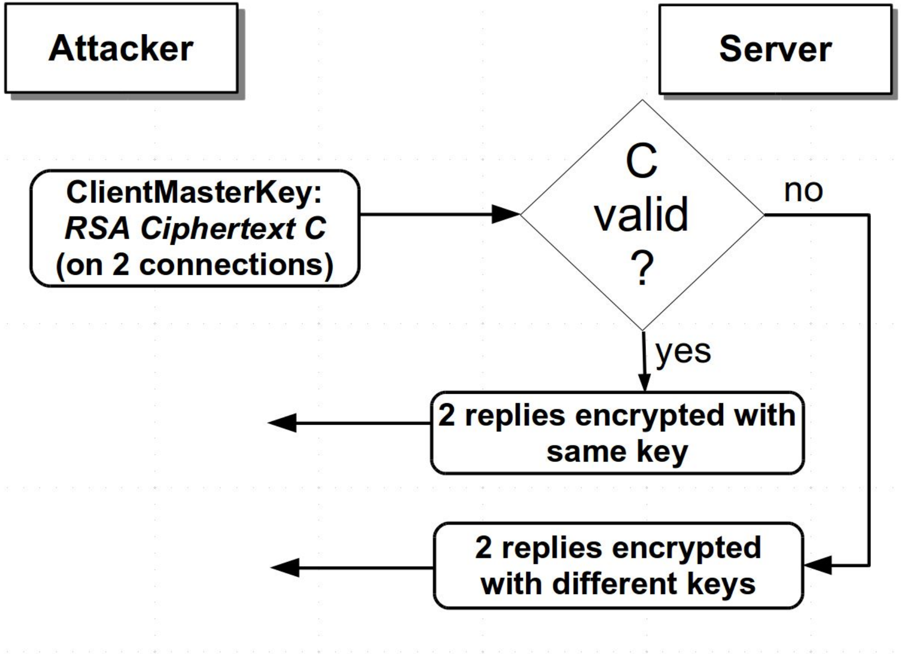
An important observation

- Attacker connects twice with same RSA ciphertext.
- Ciphertext not valid:
- 2 server replies encrypted with different keys.



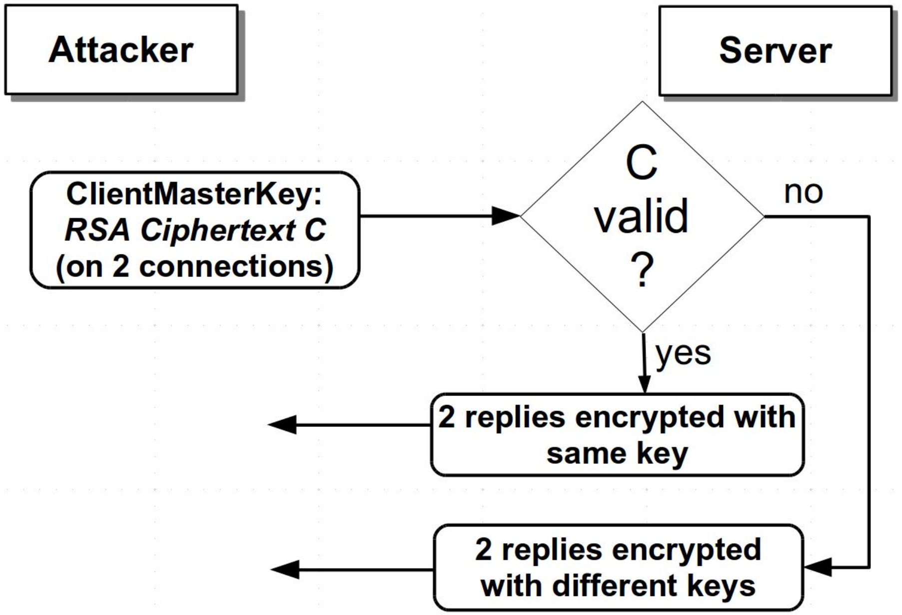
The SSLv2 RSA Decryption Oracle

- Attacker breaks 40 bit key for both messages.
- Ciphertext valid:
  - Both keys will be the unpadded RSA plaintext -> keys will be identical.



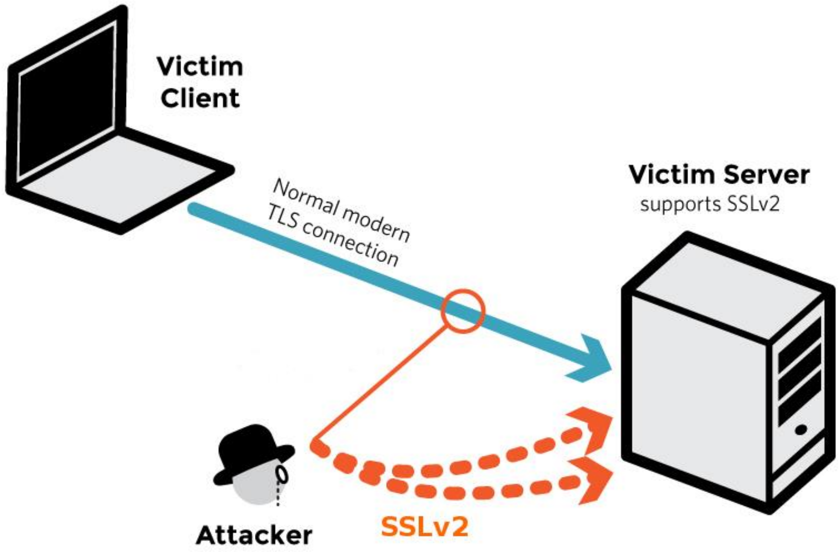
The SSLv2 RSA Decryption Oracle

- Attacker breaks 40 bit key for both messages.
- If ciphertext is invalid:
  - Both keys will be randomly generated -> keys will be different.
- Reminder: If attacker can distinguish between valid/invalid RSA message, attacker can decrypt TLS!



DROWN: Attack Outline

- Attacker records ~1,000 modern TLS connections.
- Attacker morphs TLS RSA ciphertext to SSLv2 ciphertext
  - Uses SSLv2 Bleichenbacher oracle to decrypt.
- Client never makes an SSLv2 connection.

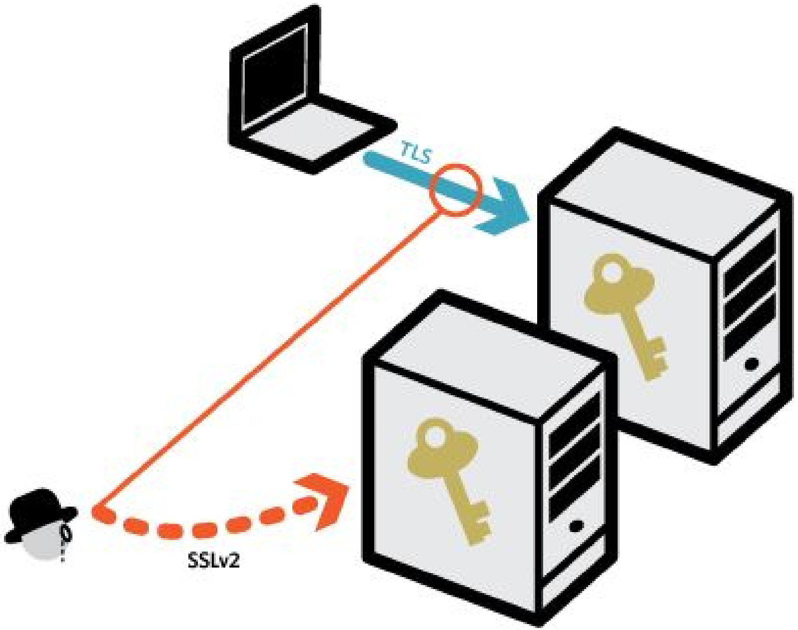


Offline work

- Attacker executes ~10K queries, breaks 40-bit key for each Bleichenbacher query.
  - 2^{50} keys tested overall.
- Feasible on modern hardware:
  - Naive CPU implementation: $21K of CPU, 114 days.
- Highly optimized GPU implementation:
  - $18K of GPUs, 18 hours, or $440 on AWS, 8 hours.
- Special DROWN: Implementation vulnerability in OpenSSL
  - 22% of trusted HTTPS servers are vulnerable
  - Negligible computation, see paper

Key reuse

- Attack HTTPS server using email server
- Widespread key reuse:
  - No protocol version in certificates
  - Certificates cost money (EV)



Impact of Key Reuse

Protocol	Port	All Certificates			Trusted Certificates		
		TLS	SSLv2	Vulnerable Key	TLS	SSLv2	Vulnerable Key
SMTP	25	3,357 K	936 K (28%)	1,666 K (50%)	1,083 K	190 K (18%)	686 K (63%)
POP3	110	4,193 K	404 K (10%)	1,764 K (42%)	1,787 K	230 K (13%)	1,031 K (58%)
IMAP	143	4,202 K	473 K (11%)	1,759 K (59%)	1,781 K	223 K (13%)	1,022 K (58%)
HTTPS	443	34,727 K	5,975 K (17%)	**11,444 K (33%)**	17,490 K	1,749 K (10%)	**3,931 K (22%)**
SMTPS	465	3,596 K	291 K (8%)	1,439 K (40%)	1,641 K	40 K (2%)	949 K (58%)
SMTP	587	3,507 K	423 K (12%)	1,464 K (40%)	1,657 K	133 K (8%)	986 K (59%)
IMAPS	993	4,315 K	853 K (20%)	1,835 K (43%)	1,909 K	260 K (14%)	1,119 K (59%)
POP3S	995	4,322 K	884 K (20%)	1,919 K (44%)	1,974 K	304 K (15%)	1,191 K (60%)

Takeaways

- Export crypto weakens modern protocols
  - Export RSA (FREAK), DH (Logjam), symmetric crypto (DROWN)
  - More weakened crypto seems ill-advised.
- Should remove obsolete crypto.
  - Long history of attacks: POODLE, Fake CA, RC4, FREAK, Logjam, Lucky 13, Sloth, ...
  - Is DROWN the last?
    - Mac-then-Encrypt, SHA-1, ...?

Thank you!

drownattack.com

Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, Yuval Shavitt

Special DROWN

- Implementation vulnerability in OpenSSL
  - because of added complexity from export ciphers.
- Present in 22% of trusted HTTPS.
- No symmetric key brute-forcing, negligible computation. Runs in a minute on a laptop.
- Allows MitM attack against DH TLS:
  - "Downgrade" the key exchange to RSA, use special DROWN to decrypt RSA ciphertext online.

QUIC

- Experimental TLS-like protocol by Google.
- 0-RTT
- Server signs a static config block, containing DH parameters, supported ciphersuites etc.
- If the client knows nothing, it prompts for the config block.
- Otherwise, it calculates shared keys and starts talking.
- Server indicates QUIC support, client will henceforth connect with QUIC
  - Can indicate support over plaintext.

QUIC MitM Attack

- Static signatures -> Forge a signature once, use it forever.
- Discovery over plaintext -> Server doesn't even support QUIC, attacker fakes support over plaintext.
- Google plans to fix both these issues.
- Attack cost with general DROWN: ~$10M.
- Attack cost with special DROWN: 2^{25} SSLv2 connections, no large computation.