draft-kario-rsa-guidance

Hubert Kario

Addressed items

- Blanket RSAES-PKCS1-v1_5 deprecation
- New improvements to timing side-channel attacks
- Recommendations about most common leakage sources
- Implicit rejection for RSAES-PKCS1-v1_5 (“Marvin workaround”)

RSAES-PKCS1-v1_5

Known to be problematic for over 25 years at this point [Bleichenbacher98]

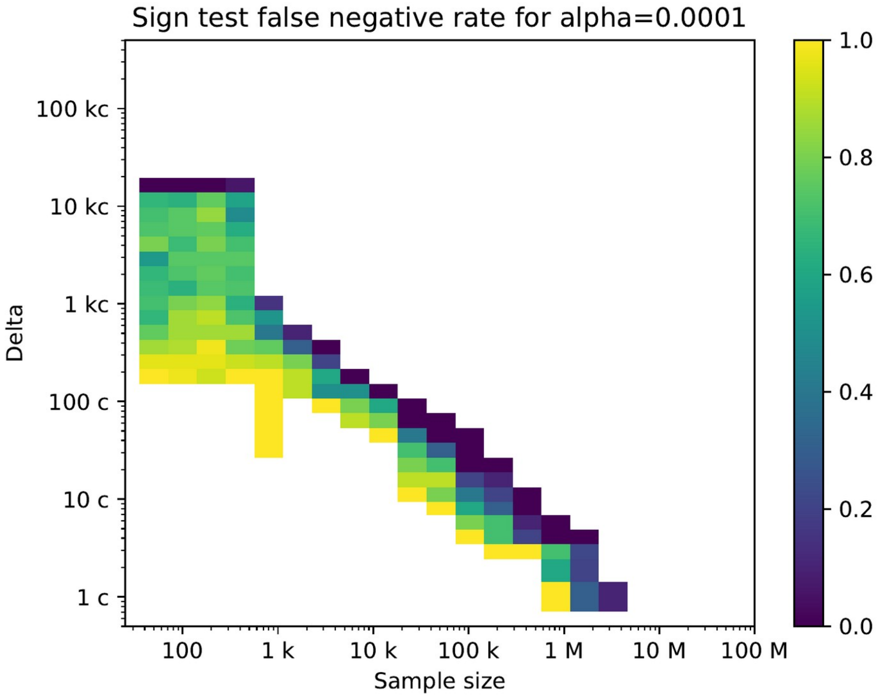
New attacks against production implementations every 3-5 years.

Already removed in TLS 1.3, WebCrypto, etc.

Improvements to timing side-channels

- The widely used Box Test [Crosby09] assumes independence in measurements
- Even if we consider just the “turbo boosting” done by modern CPUs, that assumption is fundamentally incorrect
- Given that assumptions of the statistical test are not met, it’s not reliable, therefore conclusions from it are incorrect
- That includes the “side-channels smaller than 100ns are not detectable over the network” repeated by multiple sources

Improvements to timing side-channels



Gigabit Ethernet between Xeon E3-1220 and E5-2407 v2
See “Out of the Box Testing” paper by Kario for details

Improvements to timing side-channels

- Pairwise tests (sign test, Wilcoxon signed-rank test) do *not* require the same environment for each pair
  - If the same private key is deployed on 100 machines, the attacker can attack 100 machines in parallel, combine the data and get reliable results with no additional work
- Friedman test allows for the same approach for testing multiple ciphertexts at once
- **There is no such thing as timing side-channel too small to detect over the network**

Leakage sources

- Multiple libraries (OpenSSL, NSS, ...) assumed that use of base blinding is sufficient to hide issues with use of general purpose numerical library
- Since to remove padding the value must be unblinded, the code to do that must not reveal value of high order bytes
- If numerical library uses automatic memory management it will allocate less memory to store a 1984 bit number than a 2048 bit number — this leaks value of high order bits
- 32 and 16 bit arches are particularly vulnerable, same for larger unconventional key sizes (e.g. 2049 bit)

Leakage sources

- Bad API design: rise an exception in case of padding errors
- Fine with OAEP, or signature verification, leaks precisely what the attacker wants with RSAES-PKCS1-v1_5
- Affected: M2Crypto, pyca/cryptography, python-rsa, Java, Ruby, .NET C#, Rust rsa crate, node.js, Perl Crypt::OpenSSL::RSA, etc.
- If smartcard or HSM doesn’t pad the result passed to the PC and does the PKCS#1 v1.5 depadding in hardware, it will be similarly vulnerable

Implicit rejection

- Basic idea: combine the implicit rejection from TLS with deterministic nonce generation for (EC)DSA
  - TLS generates random value, uses it in case of padding error or unexpected message size
  - (EC)DSA uses private key and message as sources of entropy to generate a deterministic nonce for signature

Implicit rejection

- In chosen ciphertext attacks on RSA the attacker doesn’t choose the whole ciphertext, they multiply it by a selected number and then pass it to the oracle
- Oracle gives information if the decrypted value (RSADP() output) starts with 0 bits or not—padding being PKCS#1 v1.5 conforming is a proxy of that
- With implicit rejection, the library returns a message always, even if padding is incorrect, thus removing the value of the remove the source of information