CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

With Great Power Come Great Side Channels:
Statistical Timing Side-Channel Analyses with Bounded Type-1 Errors

Martin Dunsche[1], Marcel Maehren[1], Nurullah Erinola[1], Robert Merget[2], Nicolai Bissantz[1], Juraj Somorovsky[3], Jörg Schwenk[1]

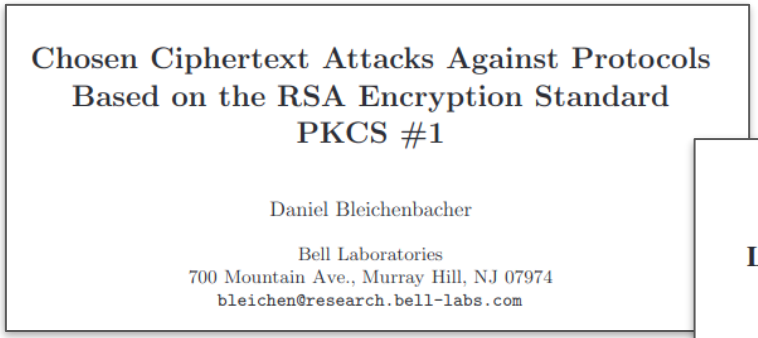
Ruhr University Bochum[1]
Technology Innovation Institute[2]
Paderborn University[3]

RUHR UNIVERSITÄT BOCHUM RUB

Gefördert durch DFG Deutsche Forschungsgemeinschaft

HGI HORST GÖRTZ INSTITUT

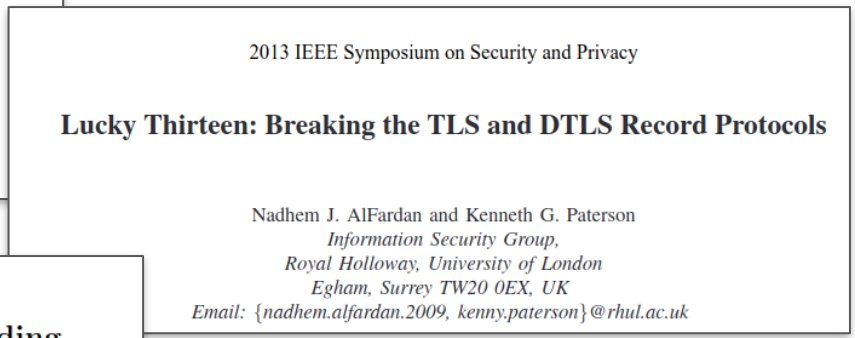
Side-Channel Attacks



Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1

Daniel Bleichenbacher

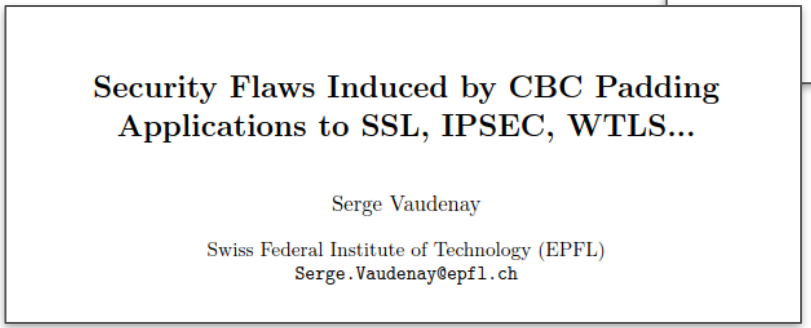
Bell Laboratories
700 Mountain Ave., Murray Hill, NJ 07974
bleichen@research.bell-labs.com



2013 IEEE Symposium on Security and Privacy

Lucky Thirteen: Breaking the TLS and DTLS Record Protocols

Nadhem J. AlFardan and Kenneth G. Paterson
Information Security Group,
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
Email: {nadhem.alfardan.2009, kenny.paterson}@rhul.ac.uk



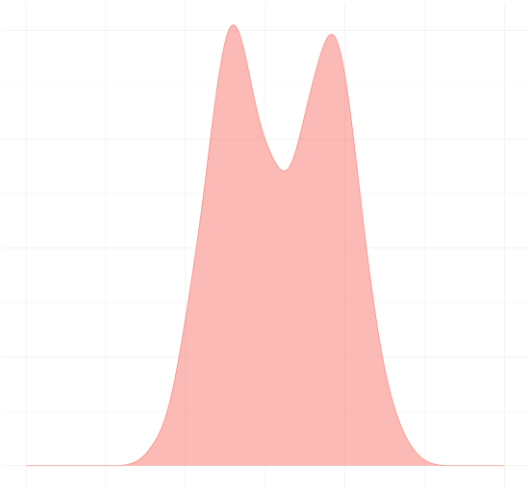
Security Flaws Induced by CBC Padding Applications to SSL, IPSEC, WTLS...

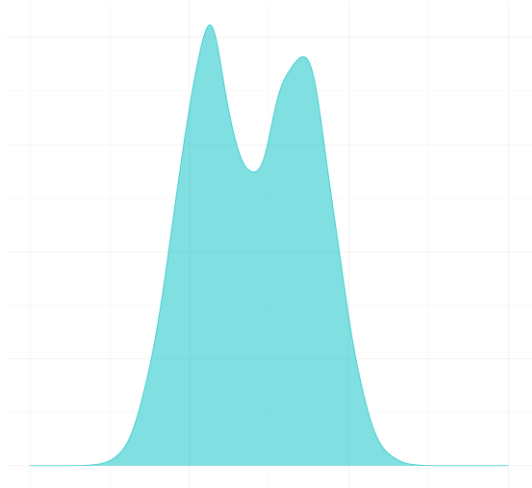
Serge Vaudenay

Swiss Federal Institute of Technology (EPFL)
Serge.Vaudenay@epfl.ch

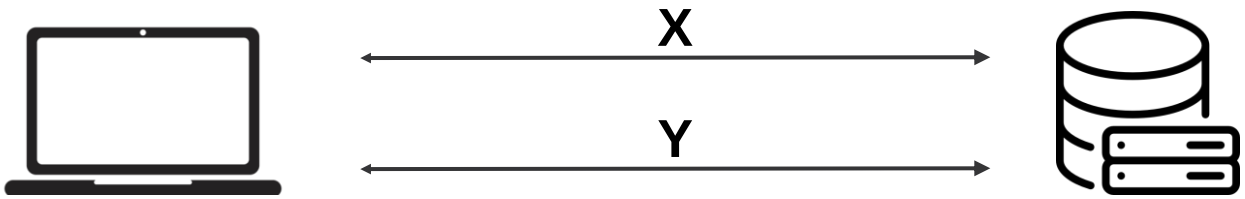
Road Towards a Statistical Test

- In cryptographic timing measurements a protocol should have secret independent execution time
- For two different inputs **X** and **Y** (e.g. padding is correct vs. padding is incorrect) we have two distributions

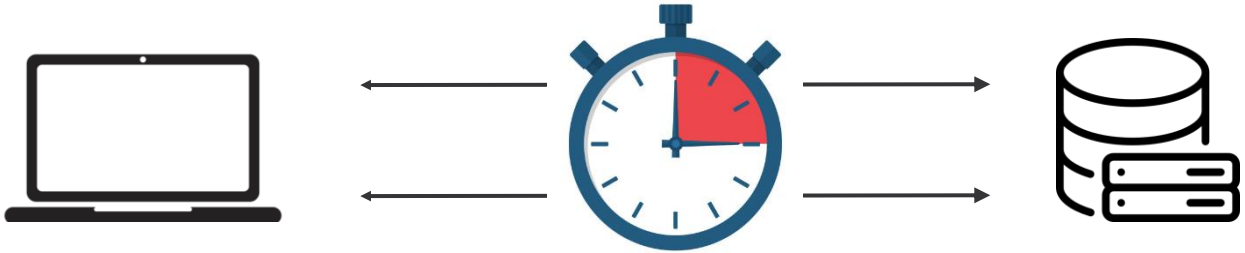




Collecting Measurements

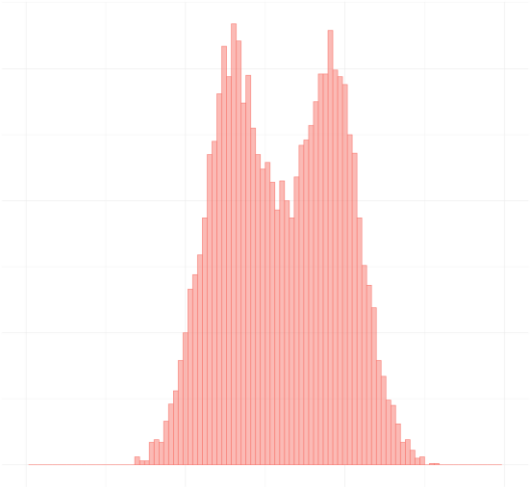


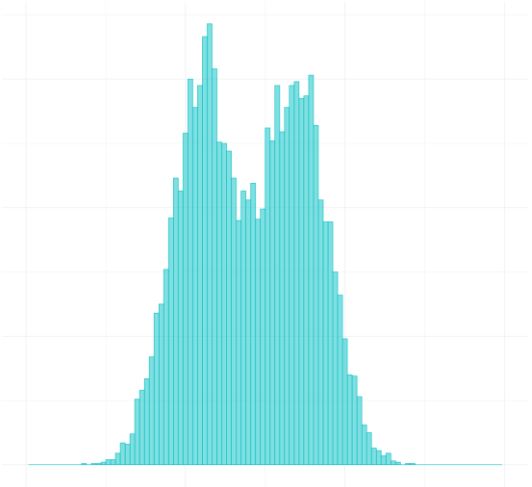
Collecting Measurements



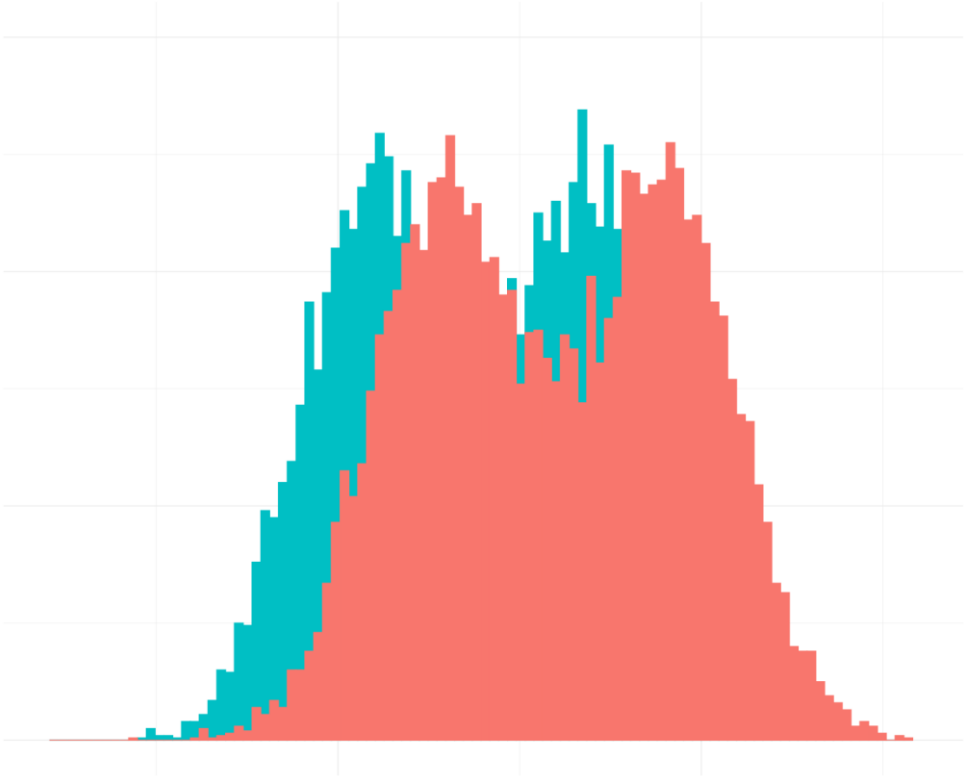
Road Towards a Statistical Test

- We collect measurements $x_1,...,x_n$ and $y_1,...,y_n$
- If the execution time is secret independent histograms should look *similar*

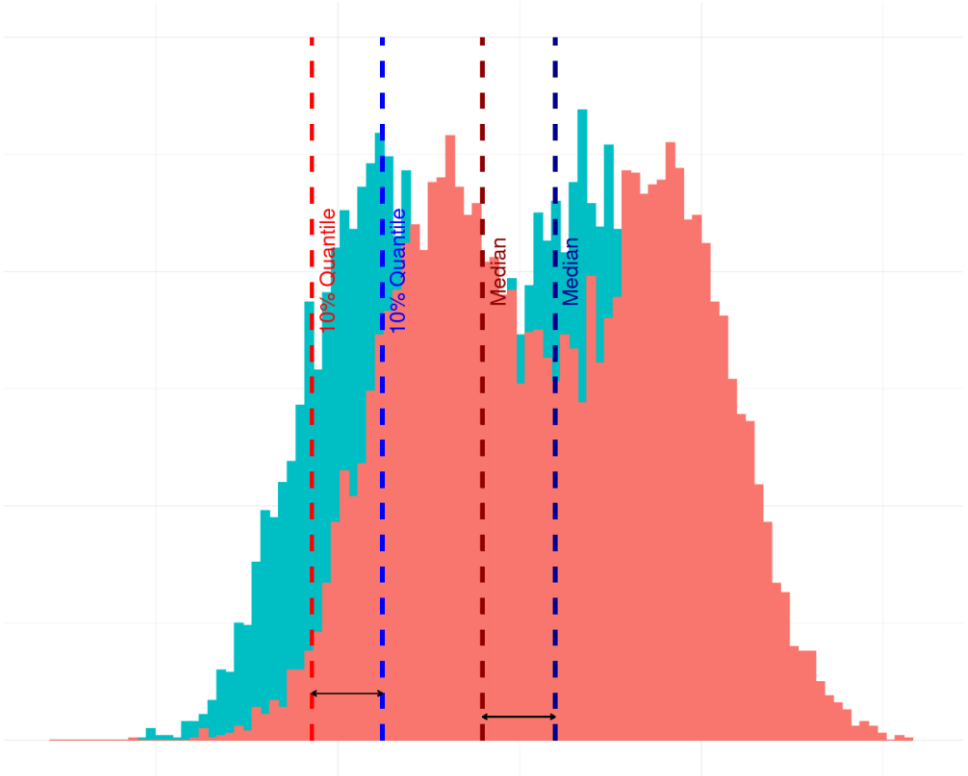




Distinguish Test Vectors X,Y



How We Distinguish Test Vectors X,Y



Deriving a Decision Rule

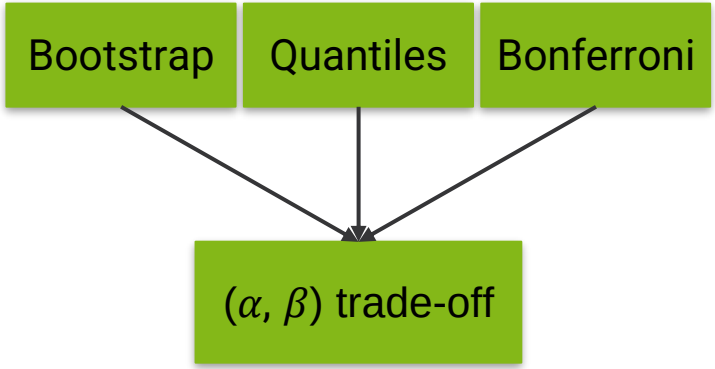
- Decision is based on data $x_1,...,x_n$ and $y_1,...,y_n$

Reality \ Decision	Side Channel	No Side Channel
Side Channel	**true positive**	**false negative**
No Side Channel	**false positive**	**true negative**

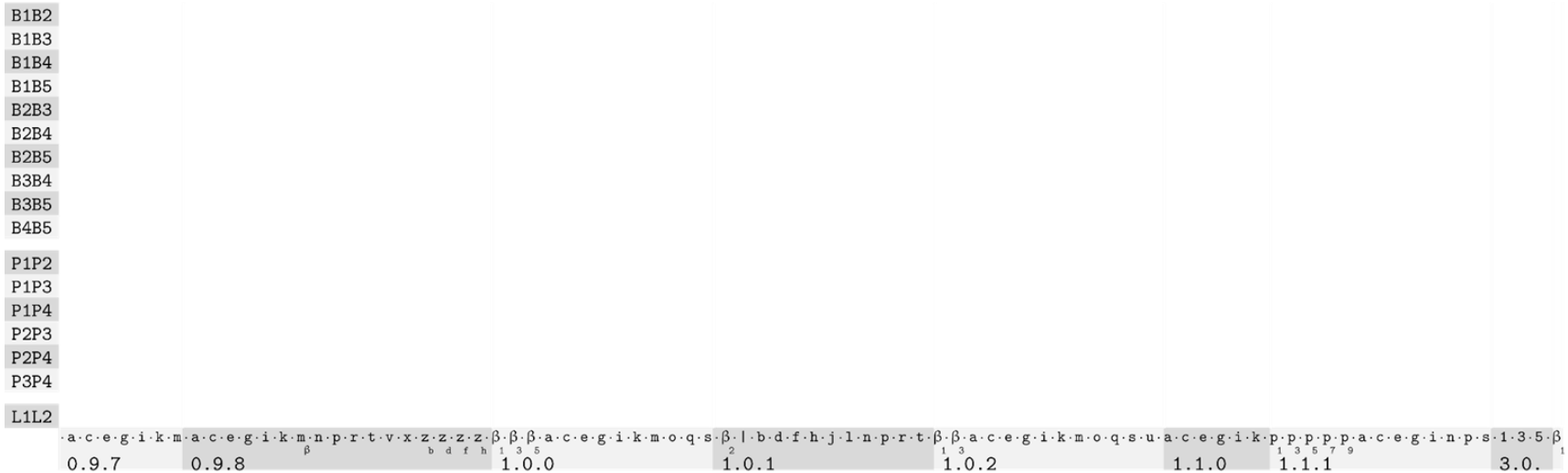
- Common approach in statistics:
  - prescribe false positive rate α
  - given α, minimize false negative rate β

R-Time-Leak-Finder (RTLF)

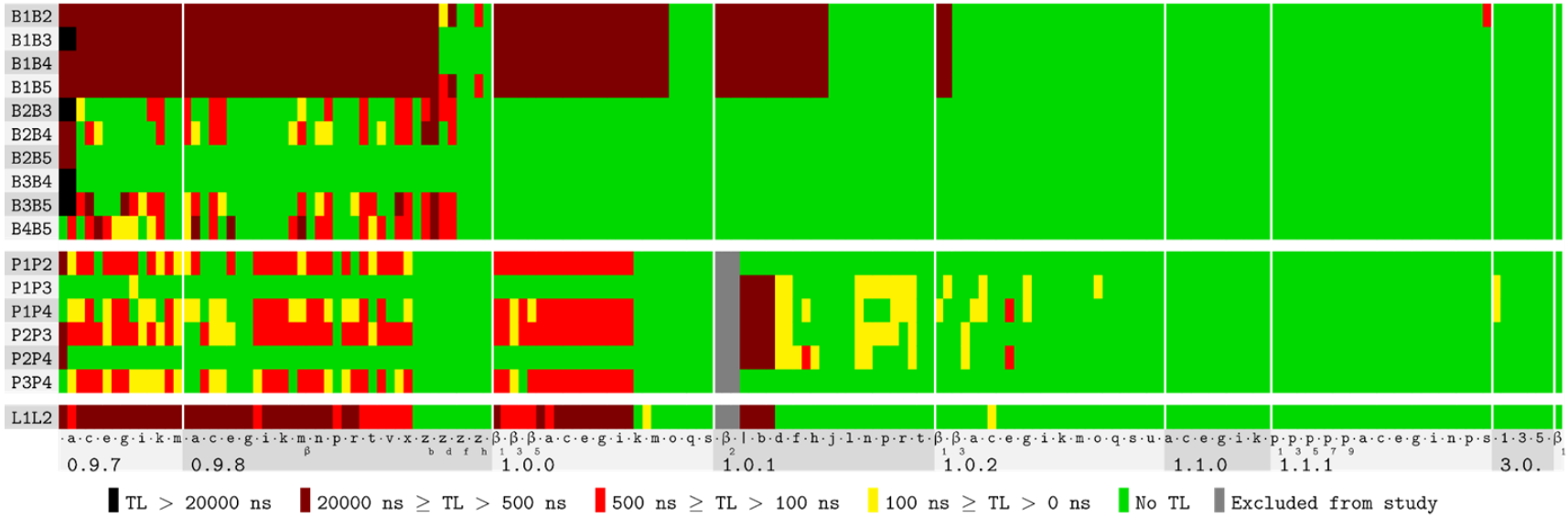
- New tool with **adequate** trade-off between false positives and false negatives
- α is an input parameter and upper bounds the false positives
- larger n decreases β
- Statistical methods we use:
  - Bootstrap (resampling): method to balance the trade-off between α and β based on the data
  - Bonferroni correction to address multiple testing



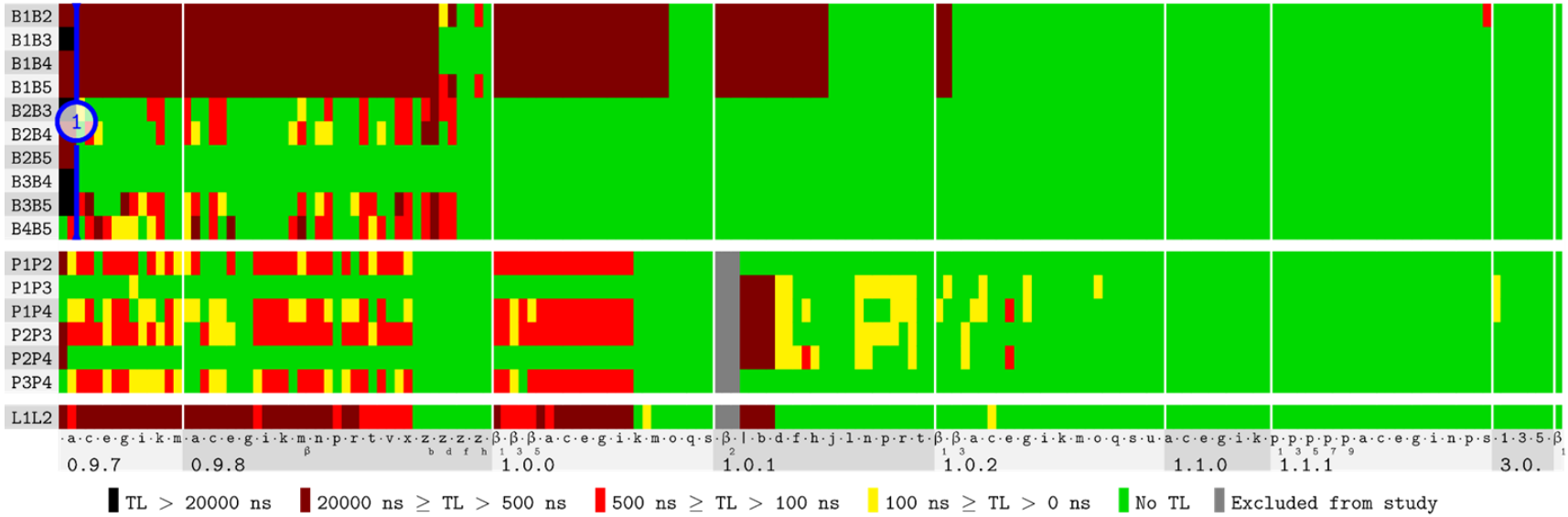
OpenSSL - α < 0.9%



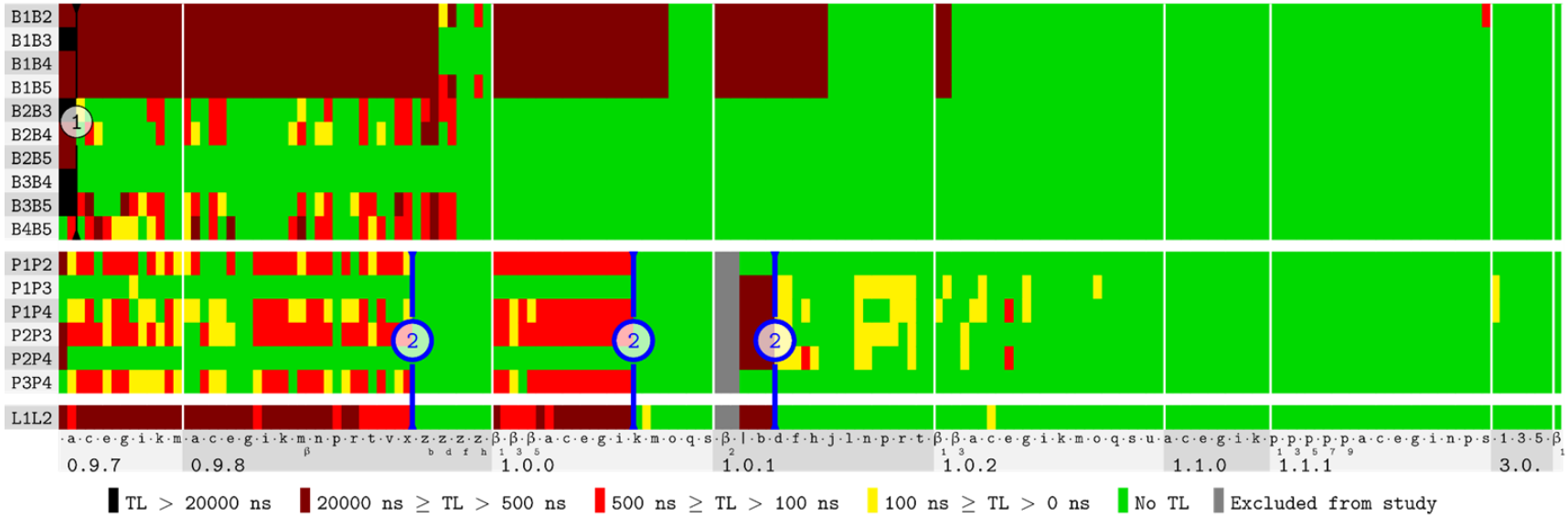
OpenSSL - $\alpha < 0.9\%$



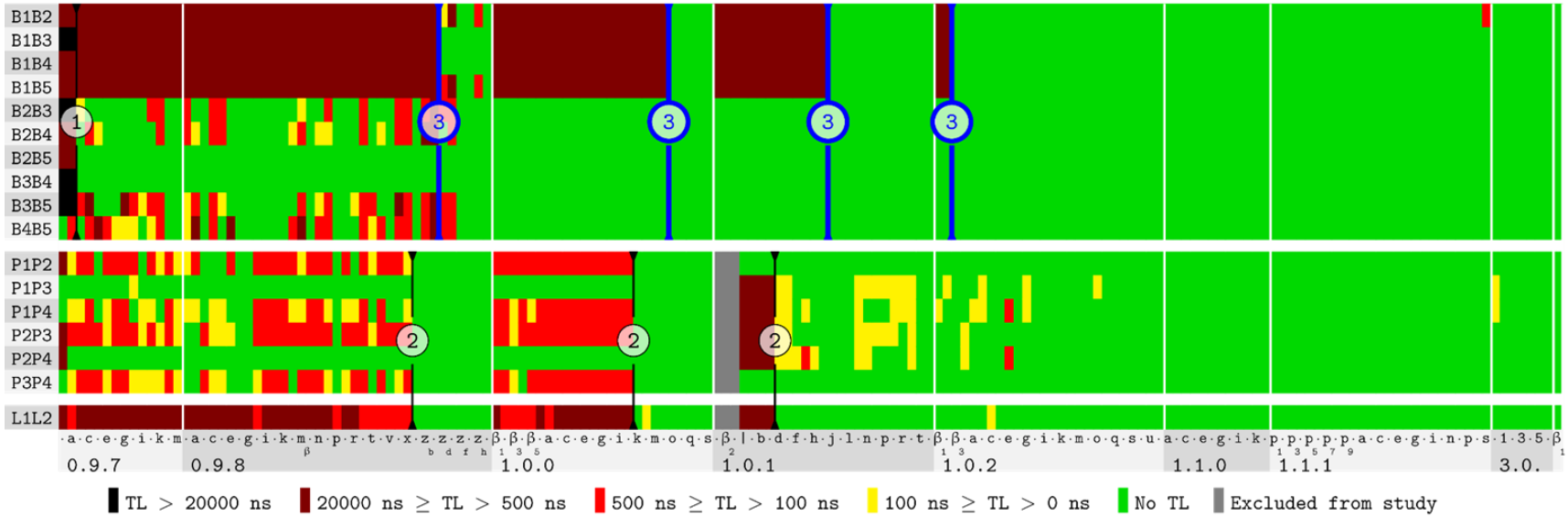
OpenSSL - $\alpha < 0.9\%$



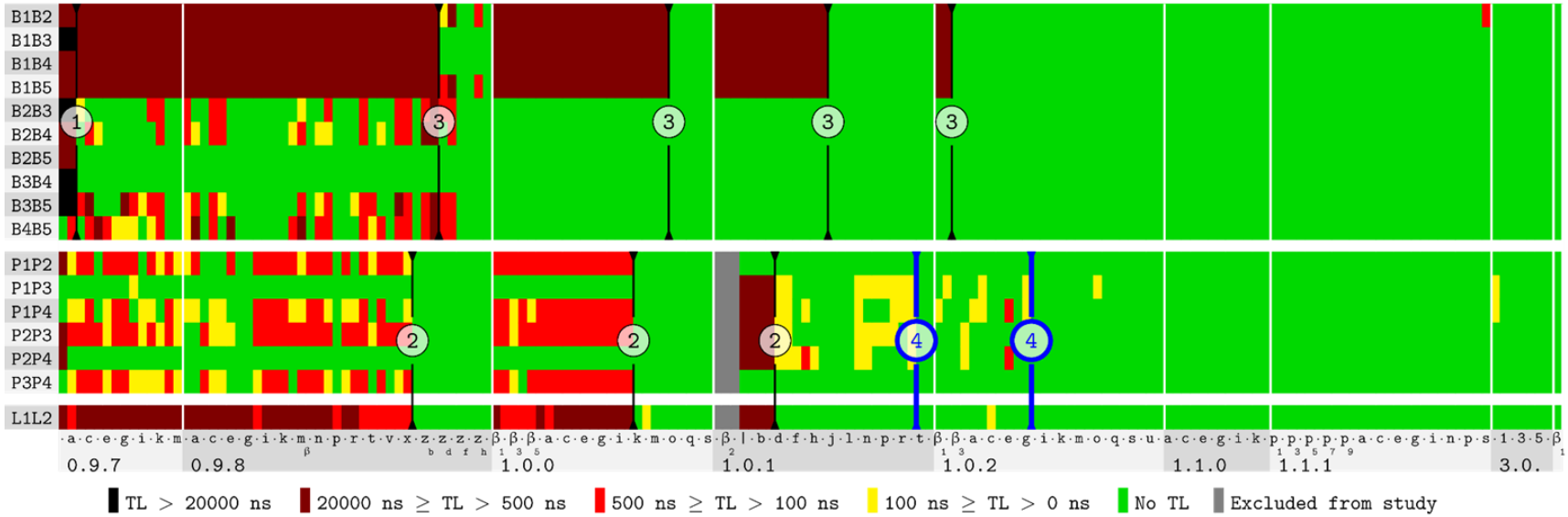
OpenSSL - $\alpha < 0.9\%$



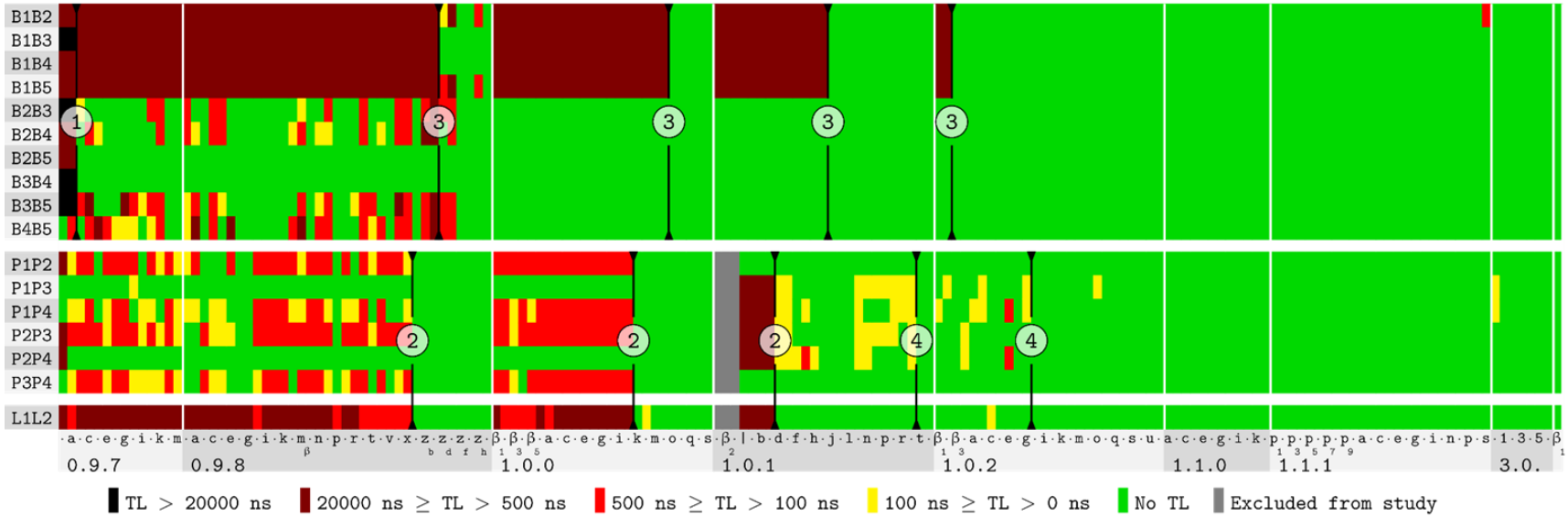
OpenSSL - $\alpha < 0.9\%$



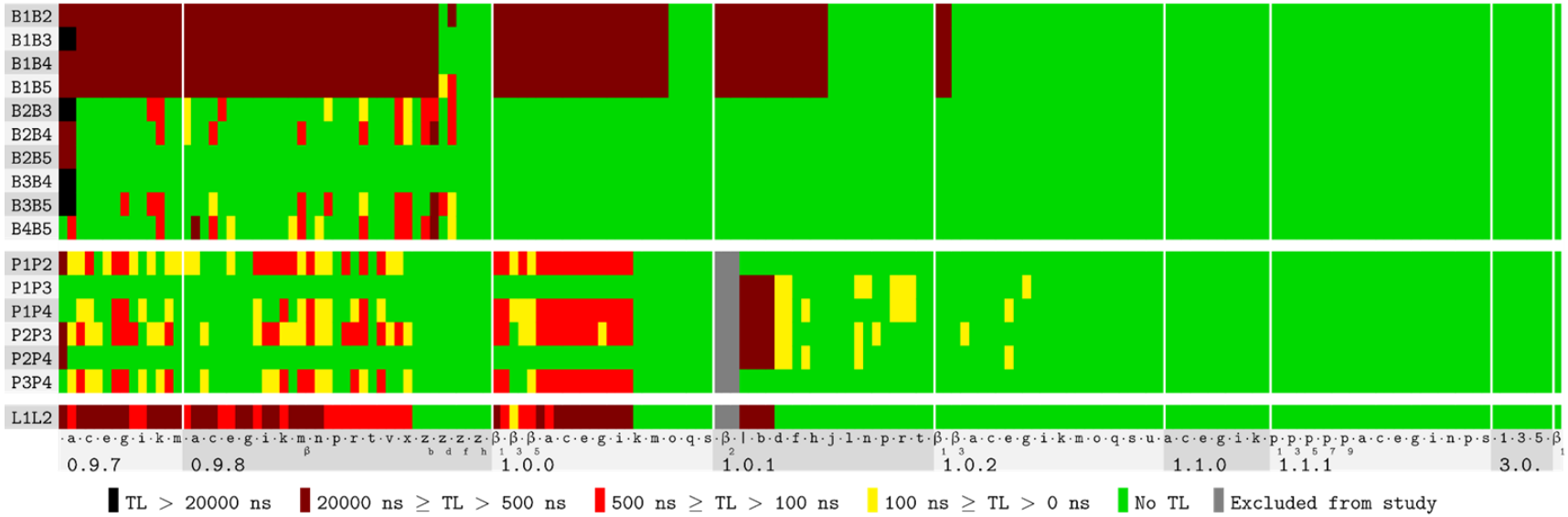
OpenSSL - $\alpha < 0.9\%$



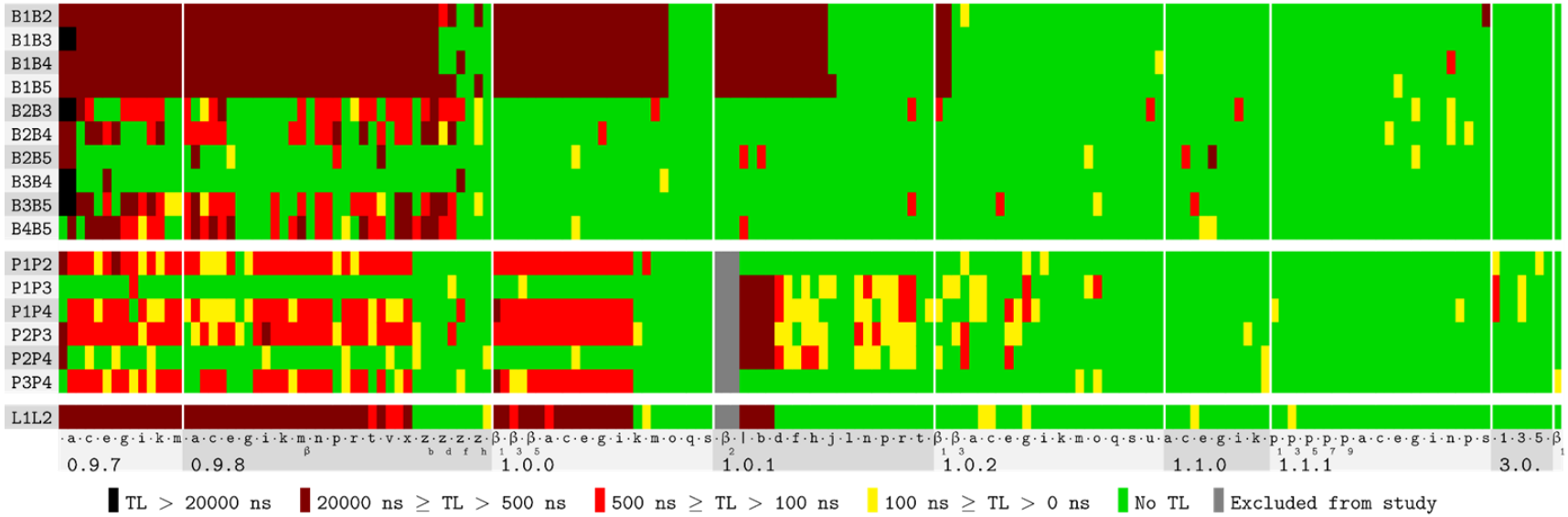
OpenSSL - $\alpha < 0.9\%$



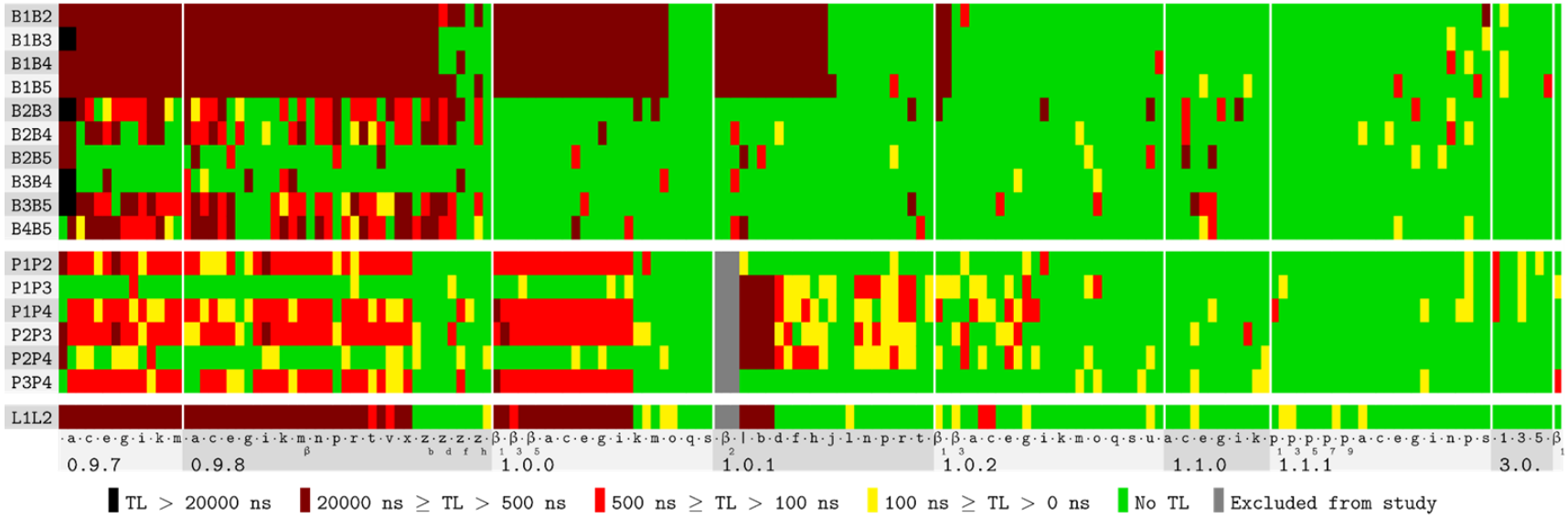
Decreasing α



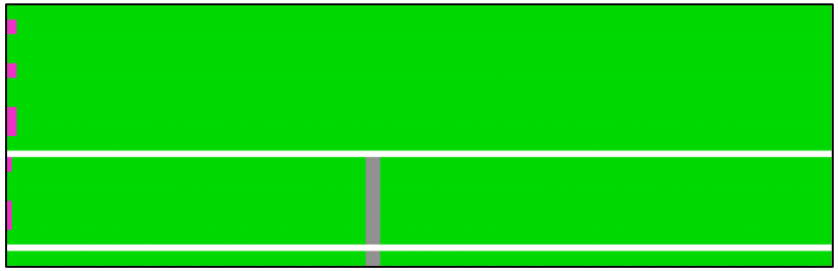
Increasing α < 9%



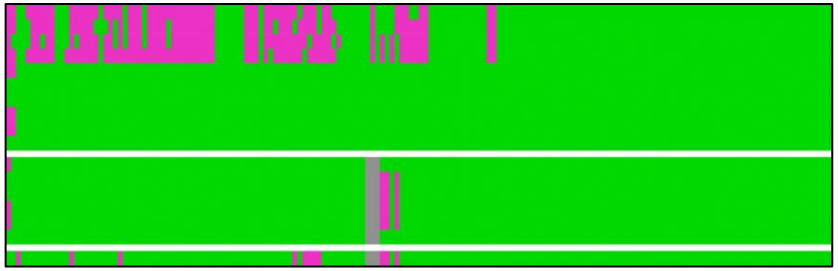
Increasing α < 18%



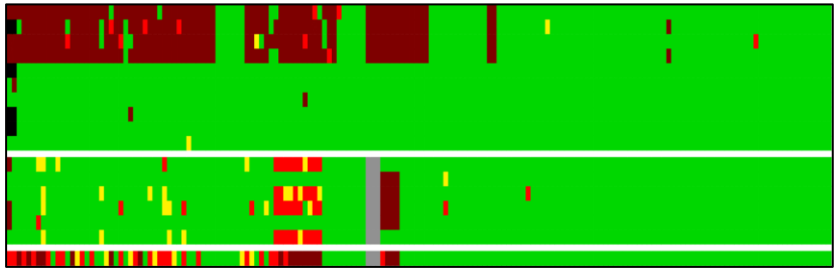
α,β-Trade Off Is Poor in *dudect*[1]



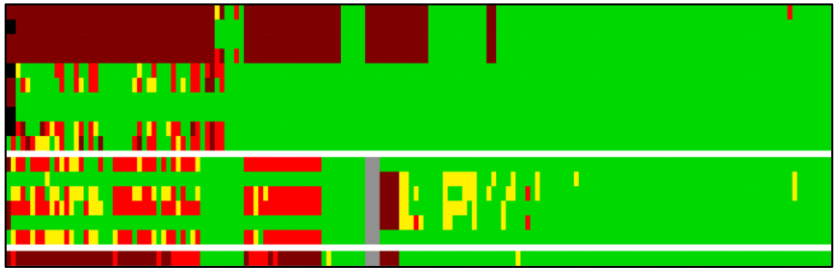
a) dudect (n=30.000, α=?)



b) dudect (n=200.000, α=?)



c) RTLF (n=30.000, α<0.9%)



d) RTLF (n=200.000, α<0.9%)

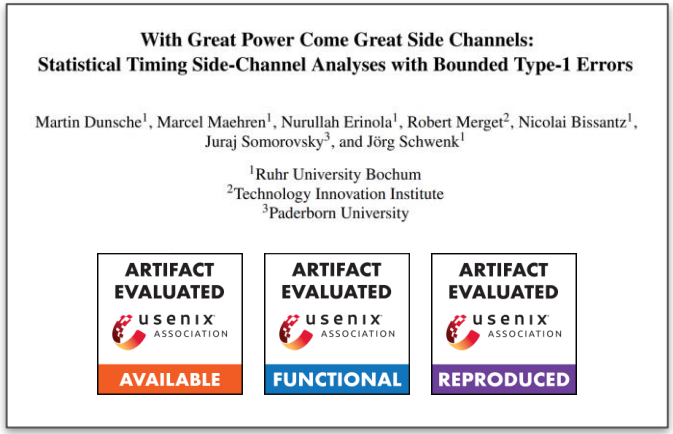
1 - *Reparaz et al. - Dude, is my code constant time? (2016)*

Summary

Statistical tool with **usable** trade-off between false positives and false negatives

Empirically outperformed existing tools

Extensive real world evaluation
- ➔ 11 TLS libraries in 823 versions
- ➔ 7 vulnerabilities in recent versions



https://github.com/tls-attacker/RTLF