Mbed TLS Tech Forum

https://github.com/Mbed-TLS

Dave Rodgman

2023-08-14

© 2022 Arm

Recent community activity (thank you!)

- Valerio Setti @Nordic
  - Active
    - driver-only ECC: Make PSA curves always a superset of ECP curves
    - Test with TF-M config and p256-m driver (part 1)
    - TLS: Clean up ECDSA dependencies
  - Merged
    - Improve outcome-analysis.sh script
    - driver-only ECC: BN.TLS testing
    - PSA maximum size macro definitions should take support into account
    - Driver-only ECC: TLS: rm uses of mbedtls_debug_print_mpi
    - driver-only ECC: BN.x509 testing
    - Define PSA_WANT_xxx_KEY_PAIR_yyy step 2/DH
    - TLS: Clean up (EC)DH dependencies
    - Backport: crypto_config_test_driver_extension: handle PUBLIC_KEY the same way as KEY_PAIRs
- Tomi Fontanilles @Nordic
  - Implement non-PSA pk_sign_ext()
- Kusumit Ghoderao, Saketh Sunkishala @ Silicon Labs
  - PBKDF2 CMAC implementation
  - Fix IAR enum conversion warnings when using mbedtls_md_type_t and mbedtls_cipher_type_t
- Misc
  - Fix a few unchecked return values - Chien Wong
  - Fixed x509 certificate generation to conform to RFCs when using ECC key - marekjansta
  - Backport 2.28: Fixed x509 certificate generation to conform to RFCs when using ECC key - marekjansta
  - rsa_signature: Use heap memory to allocate DER encoded RSA private key - Sarvesh Bodakhe
  - asn1parse: Require minimal-length encodings of lengths – Demi Marie Obenaur
- Pol Henarejos – SHA3 / EdDSA
  - XChaCha20 and XChaCha20-Poly1305 support
  - Add support to Ed448 in EdDSA
  - Add support for SHA-3 KMAC
  - SHA-3 cSHAKE128 and cSHAKE256 support
  - SHA-3 SHAKE128 and SHAKE256 support
  - Add support for EdDSA with ed25519 curve (pure ed25519, ed25519ctx and ed25519ph)

© 2022 Arm

Major activities within core team

https://github.com/orgs/Mbed-TLS/projects/1

- Planning Mbed TLS 3.5 - September – October 2023
  - Size optimization (including driver-only ECP, bignum)
  - p-256m – reduce code size for SECP256R1 ECDH and ECDSA
- Planning Mbed TLS 3.6 LTS - end of 2023 (maybe early 2024)
  - TLS 1.3 early data, record size limit
  - PSA multi-threading support
  - Accessor functions for fields made private in 3.0
  - Driver-only cipher and AEAD
- Planning Mbed TLS 4.0 – mid 2024?
  - PSA_CRYPTO_C / CLIENT always on
  - Consume PSA-Crypto repository as source of PSA and crypto code
  - Remove some legacy interfaces & features
- PSA Crypto – prototyping move to separate repository
- Size optimization
  - This is the primary focus for Mbed TLS 3.5
  - Expect driver-only to deliver good improvement here
- CI
  - Testing on Arm coming soon
- Review workload
  - Struggling for review bandwidth – any assistance from the community is hugely valuable
  - Easing the general review load accelerates progress on work prioritized by the community
  - Increased use of draft PRs

© 2022 Arm