DHBW.Krypto-MSA

Beispiel anhand der Nachricht "morpheus"

Encrypt/Decrypt with RSA
`encrypt message "morpheus" using RSA and keyfile rsa32`
`decrypt message "JbkPFt+y+j8=" using RSA and keyfile rsa32`
`crack encrypted message "JbkPFt+y+j8=" using RSA and keyfile rsa32`
Cracking fails due to timeout

Crackable RSA example
`crack encrypted message "BqGfopSO" using RSA and keyfile rsa24`

Encrypt/Decrypt with Shift
`encrypt message "morpheus" using shift and keyfile shift13`
`decrypt message "zbecurhf" using shift and keyfile shift13`
`crack encrypted message "zbecurhf" using shift`

Anmerkungen zur Implementierung

Um die Anwendung funktionsfähig zu bekommen, sind Abweichungen von der Aufgabenstellung notwendig.

Eventbus
Damit der Eventbus korrekt funktioniert, musste die `failureaccess-1.0.1.jar` ergänzt werden, da sonst immer nur Fehlermeldungen angezeigt wurden.

Cracker
Der RSA Cracker benötigt eine `File` für den öffentlichen Schlüssel. Um beim Arbeiten mit der Komponente keine Fallunterscheidung vornehmen zu müssen, wurde das Interface für den Shift Cracker auch um `File` erweitert. Das übergebene Objekt wird jedoch nicht verarbeitet.

Schnittstelle RSA Cracker:
`String decrypt(String encryptedMessage, File publicKeyfile);`

Schnittstelle Shift Cracker:
`String decrypt(String encryptedMessage, File file);`

Der CQL Befehl wurde erweitert mit der Option eine Datei zu übergeben: `crack encrypted message "[message]" using [algorithm] [and keyfile [filename]]`