

# **SECURE CHANNEL PROTOCOL BETWEEN JAVA CARD AND PC APPLICATION**

Milan Patnaik

Mayank Samadhiya

Suresh Baddipudi

Mohammad Akhtar

# OUTLINE

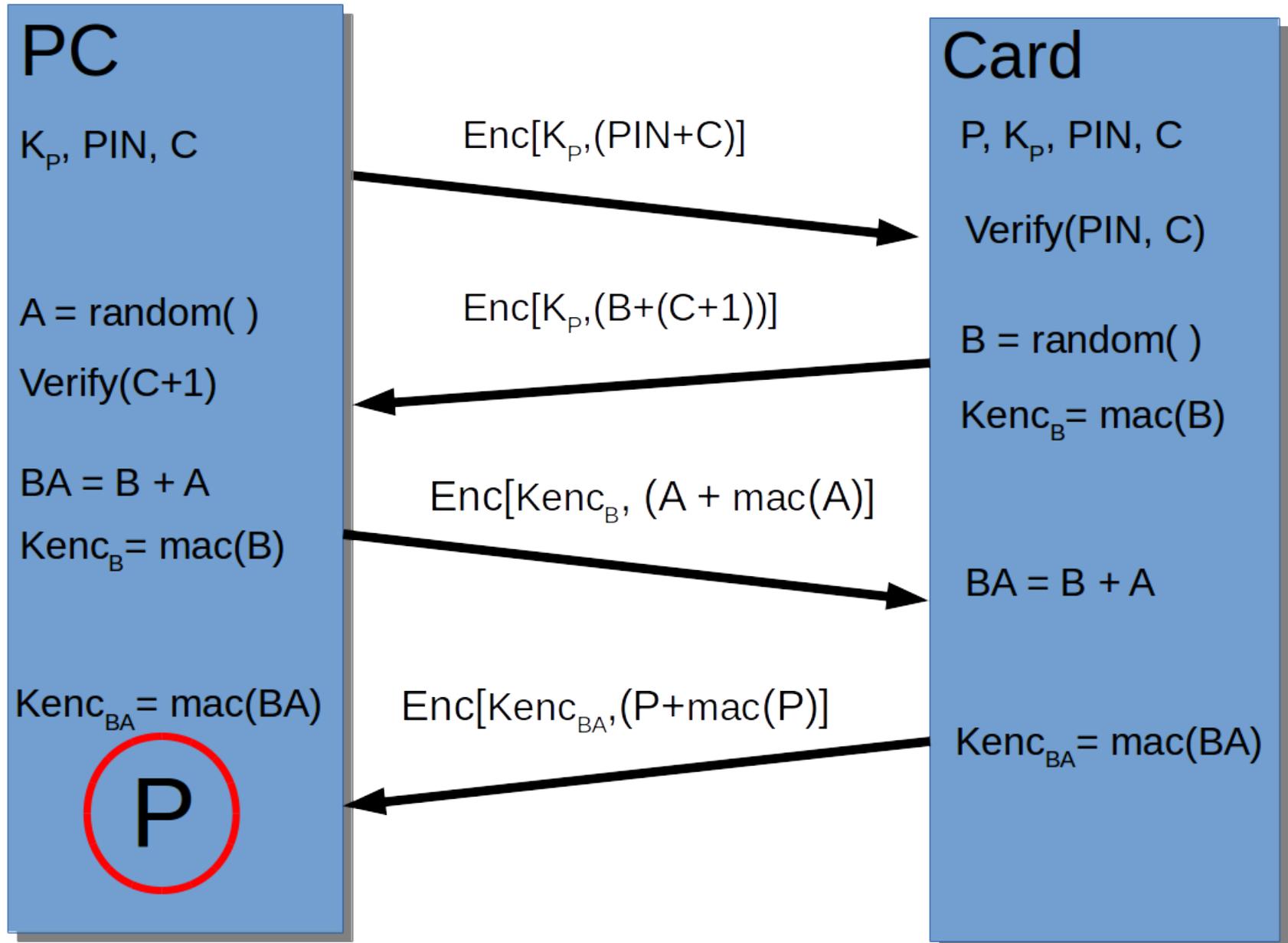
- Specification.
- Secure Channel Protocols.
  - SymSec Protocol.
  - AsymSec Protocol.
- Implementation.
- Observations.
- Future Work.

# SPECIFICATION

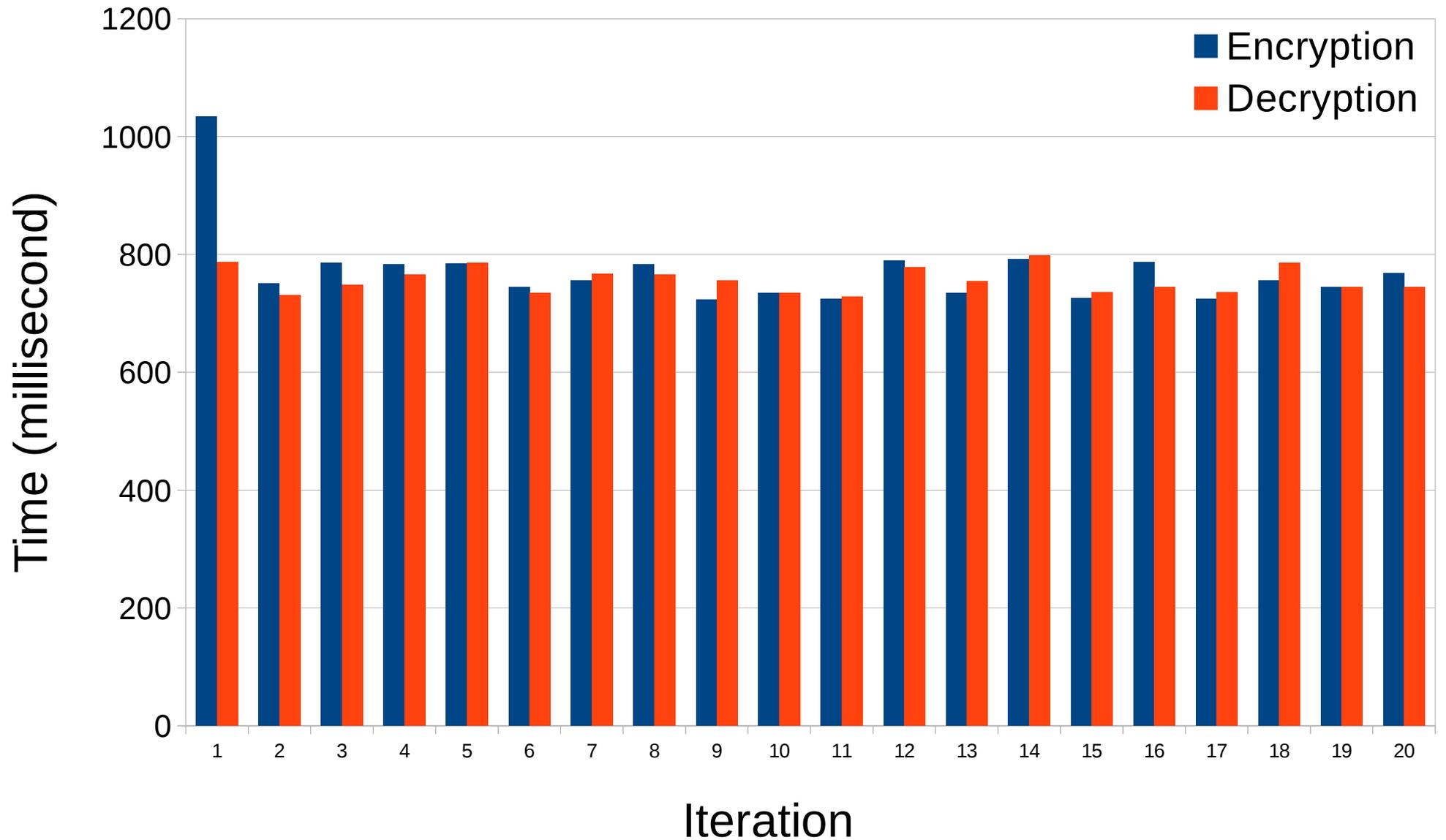
- Propose a **Symmetric Key Cryptography** based **secure channel protocol** between Java Card and PC application for File Encryption & Decryption.
- Propose an **Asymmetric Key Cryptography** based **secure channel protocol** between Java Card and PC application.
- **Compare the timings for file encryption and decryption** using the above two secure channel protocols.

# SECURE CHANNEL PROTOCOL

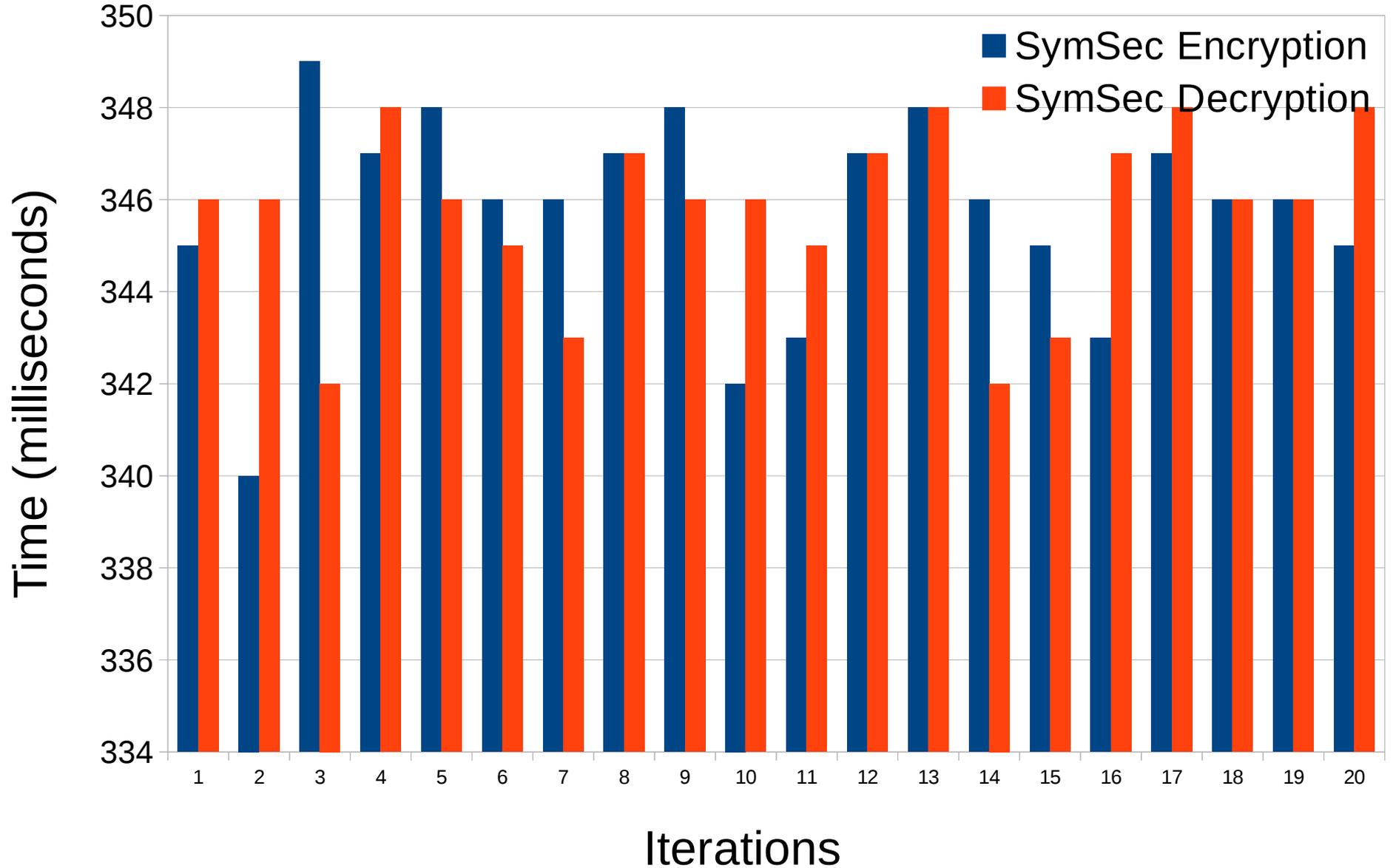
- SymSec Protocol



# TIME : ENCRYPTION & DECRYPTION

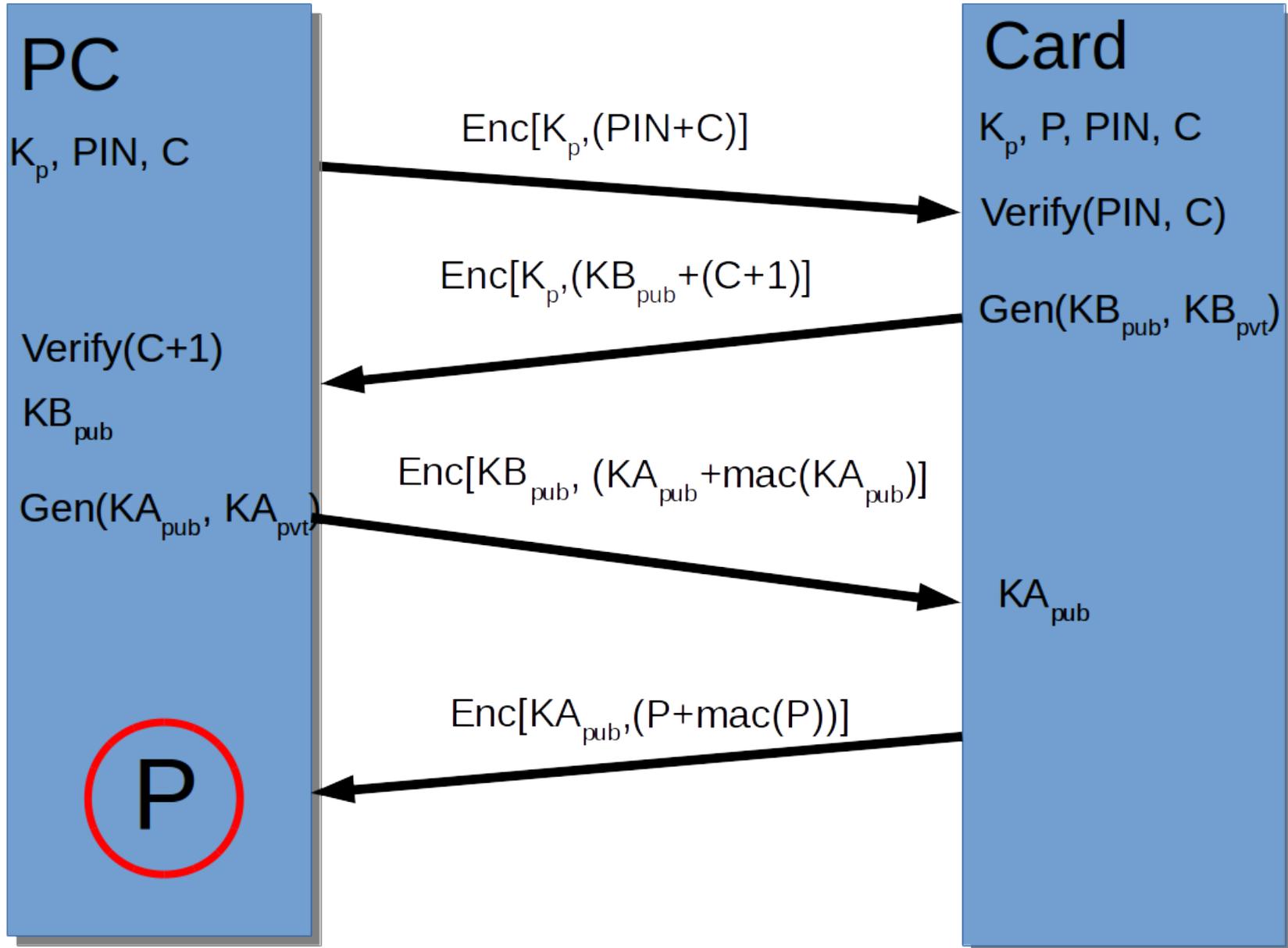


# TIME : SYMSEC



# SECURE CHANNEL PROTOCOL

- AsymSec Protocol



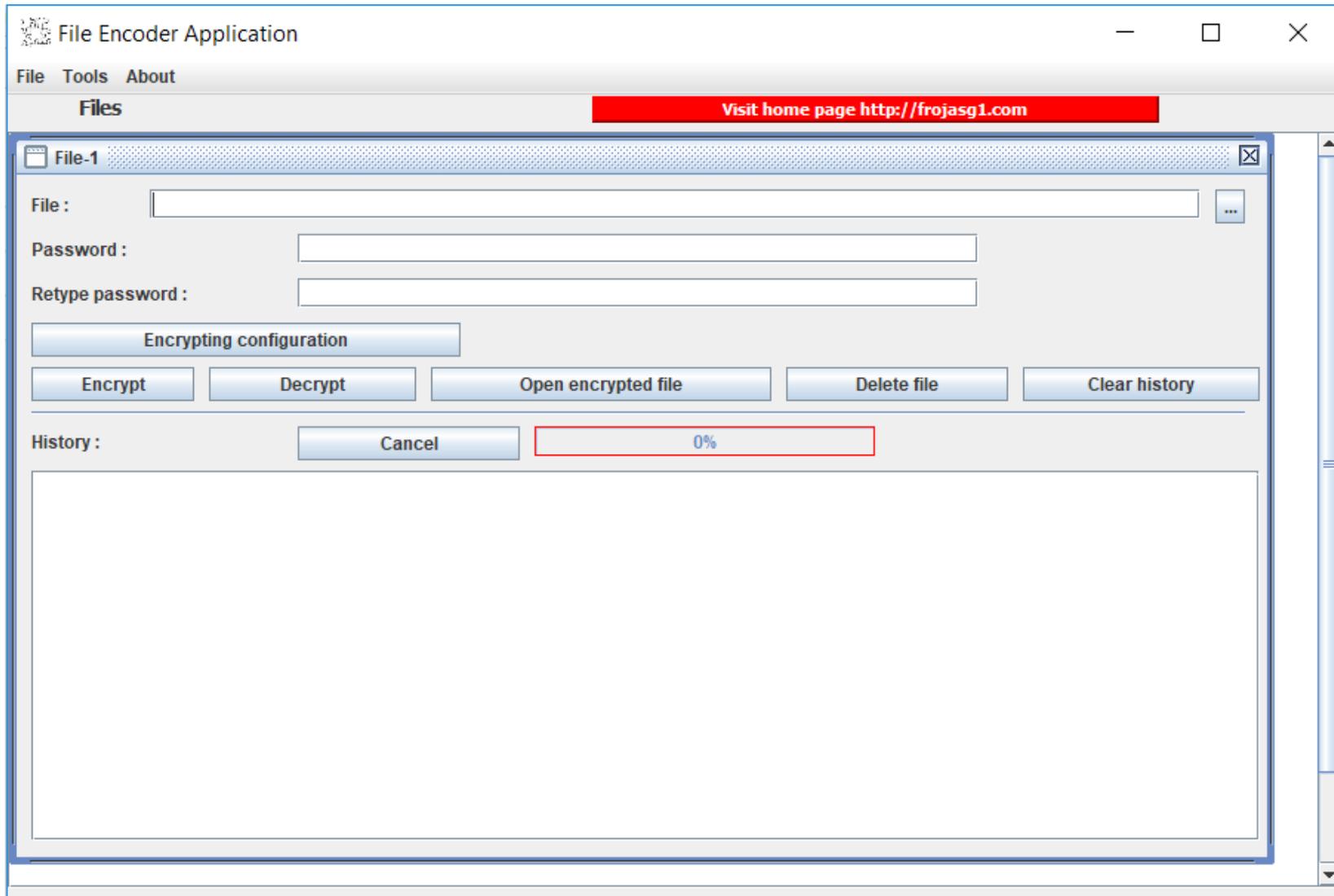
# IMPLEMENTATION

- File Encoder Application (FEApp) for file encryption and decryption using user passwords[1].
- Applet Code.
  - Time : 350 ms.
- SimpleAPDU class added to FEApp.
- Setup Tasks.
  - Setting of PIN.
  - Setting Pre-Shared Key (AES-256).
  - Setting of Passphrase.
- User Tasks.
  - Password extracted from card via Secure Channel Protocol.
  - File encryption and decryption.

[1] <https://github.com/mayanksamadhiya/SmartCrypto>

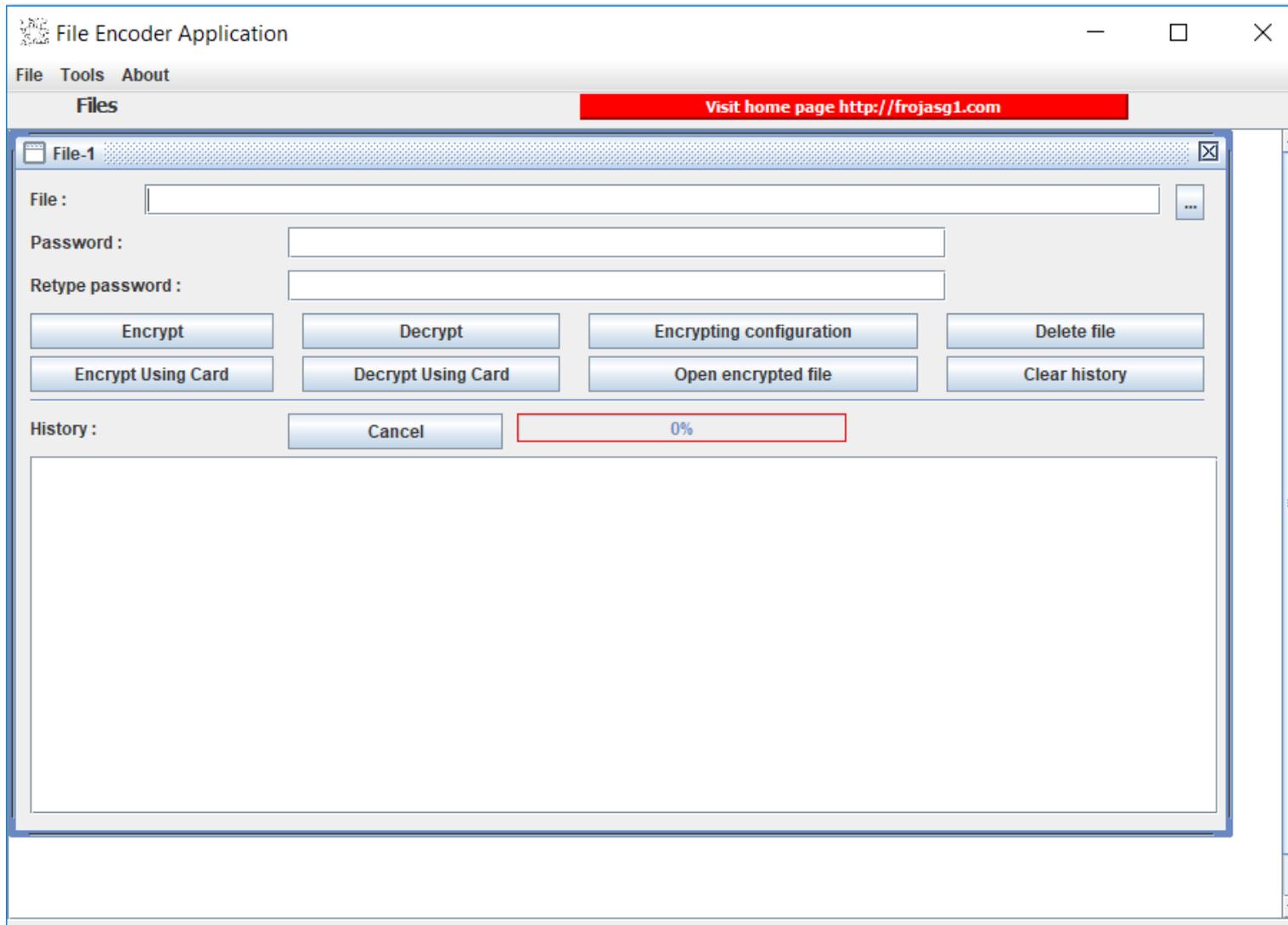
# IMPLEMENTATION

- Original Application : : File\_Encoder\_v1.1



# IMPLEMENTATION

- Java Card Application : File\_Encoder\_v2.0



# OBSERVATIONS

- Implementation of Deffie Hellman using RSA encryption.
- Criticality of transient memory space.
- Simplicity vs Security.
- Java Card versions and supported algorithms.
- Exporting Public Key.
  - Java `<publickey>.getModulus` returns 129 bytes.
  - Java `<publickey>.setModulus` of 129 bytes fail in Card.
  - First byte is sign byte.
- Padding.

# OBSERVATIONS

- Speed.
  - RAM Vs EEPROM.
  - API Algorithms.
  - Resource allocation.
  - Initialization of Cipher and Signature.
  - Garbage collection.
- Security.
  - API Algorithms.
  - Session data.
  - PINs and Pre-Shared Keys.
  - Resource allocation in constructors.

# DISTRIBUTION OF WORK

- Design of SymSec Protocol.
  - Milan & Akhtar.
- Design of AsymSec Protocol.
  - Mayank & Suresh.
- Implementation of SymSec Protocol.
  - Milan & Akhtar.
- Implementation of AsymSec Protocol.
  - Suresh & Mayank.
- Integration with FEApp GUI.
  - Suresh & Milan.
- Report and Presentation.
  - All.

# FUTURE WORK

- Completion of AsymSec Protocol.
- Compare time of file encryption and decryption by SymSec and Asymsec Protocols.