Use of Elliptic Curve Crypto (ECC) for next generation CEPAS

(for CEPAS Review Task Force consideration)

Mr. LIN YIH

email: dartpl@singnet.com.sg

Director, Digital Applied Research and Technology Pte Ltd

3 Science Park Drive #01-03/04

Tel: 67787969, 67787684

iTSC

an industry partnership supported by SPRING Singapore and IDA

Foreword

- This is a “gentle tutorial” on use of ECC for authentication and signature
- By design, CEPAS credit and debit operations require authentication
- By design, CEPAS transactions are cryptographically signed by card
- ECC can be useful for future CEPAS

Why ECC ?

- Reduce the damage caused by losing a SAM that holds the system wide ***secret*** key
- Store only public key in the SAM
- Or even totally eliminate the use of SAM

Why ECC and not RSA?

- ECC parameters are more compact – less data to store and transmit
- ECC-163 is equivalent to RSA-1024 (widely accepted statement)
- 30msec ECC-163 Sign operation (source: NXP SmartMX brochure)
- ECC allows secure communication (EC Diffie-Hellman key exchange)

How to use ECC for authentication

- There are many methods / protocols usable for authentication (e.g. Schnorr, Okamoto, etc.)
- The easiest to understand is based on Sign and Verify
- Ask the reader (or card) to Sign a random number (using its private key), Verify that signature using its verifiable public key)
- Note: Sign and Verify are cryptographic

New CEPAS Debit protocol...1

Basic setup *for card to verify reader*:

- Reader stores a reader private key (disallow read)
- Reader stores a reader public key, digitally signed by the Issuer
- Public key + signature = public key cert
- Card stores the Issuer public key

New CEPAS Debit protocol...2

Steps:

- Reader sends a Pre-Debit command, giving the reader public key cert to the card
- Card verifies the reader public key cert, by using its stored copy of issuer public key
- If verification is successful, stores the reader public key in memory, and returns a card random number to reader
- * This is like “GetChallenge”

New CEPAS Debit protocol...3

Steps:

- Reader signs the card random number using its internal reader private key
- Reader sends the signed random number, plus other debit parameters, to the card using a Debit command
- Card verifies the signed random number, using reader public key obtained from Pre-Debit

New CEPAS Debit protocol...4

Steps:

- If the verification is successful, the card proceeds with debit
- Subsequently, card should produce a signed "receipt" to the reader
- Hence card should also store a transaction signing private key, but details omitted here to avoid confusing authentication with transaction signing

Implementation Notes

- Every reader can have its own random private key – NOT based on diversification from a secret master key
- Similarly, every card can have a random private key – NOT based on diversification from a secret master key
- The storage of these public or private keys in a database is ***optional***
- What is important is the Issuer Private Key

State of ECC in the market...1

- ECC is considered “light weight” hence it is well known in areas such as: wireless sensor networks, RFID, etc.
- It is also specified as an authentication method in e-passport “Active Authentication using ECC”
- There are many open source libraries on the internet (suitable for ARM, micro-controllers, PCs)

State of ECC in the market...2

- Nevertheless, there is a learning curve involved, for IT professionals who are only familiar with symmetric secret key cryptographic systems
- Of course, existing equipment must be upgraded – but this is also required if CEPAS were to upgrade from 3DES to AES-256

END OF PRESENTATION
THANK YOU!

OPEN FOR Q&A