USC Viterbi
School of Engineering

Secure Hardware Extensions

Tanishq Ashish Kothari

Tom Xu

DATE- 09/21

What is SHE?

- It is an on chip extension given to a micro-controller.
- Embedded hardware in ECU.
- Works as secured storage for the keys.
- Supports basic symmetric primitive (AES).
- Follows crypto principles of key storage - Authenticity, Integrity and Confidentiality.

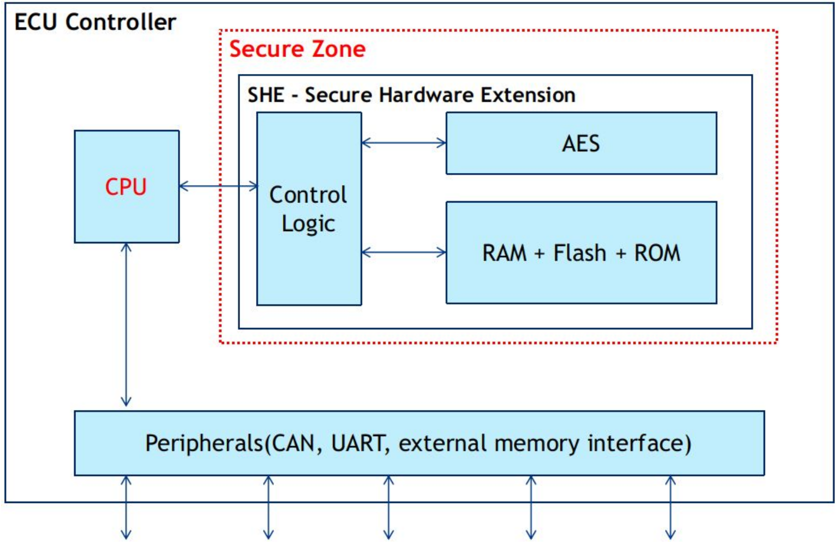
SHE - Objectives

- To provide the protection of cryptographic keys from software attacks
- Provide an authentic software environment
- Allow for distributed key ownerships
- Security depend solely on the underlying algorithm and the key confidentiality
- Provides a secure environment for its components

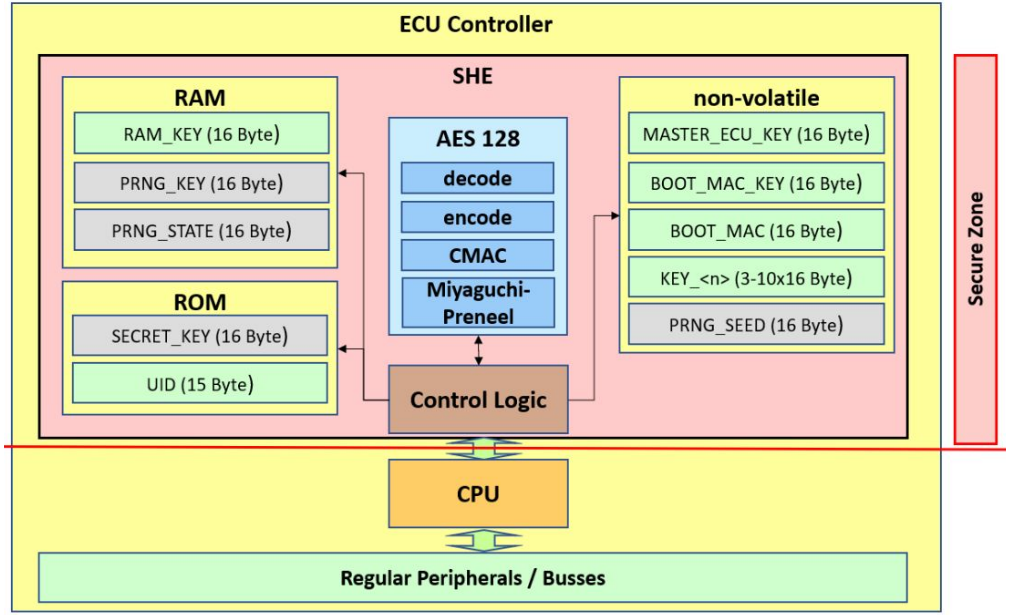
SHE Requirements

- Should be known as on chip extensions for MCU.
- Peripheral integration to CPU must done so that there shouldn’t be an external modification of data.
- Should not have external connections other than AUTOSAR specified.
- Status needs to be notified via signals.

Simplified logical structure


ECU Controller
Secure Zone
SHE - Secure Hardware Extension
CPU
Control Logic
AES
RAM + Flash + ROM
Peripherals(CAN, UART, external memory interface)

Detailed logical structure



SHE and HSM

HSM is similar to SHE when it comes to the protection of cryptographic keys, but in SHE there are specifications provided for the primitives.

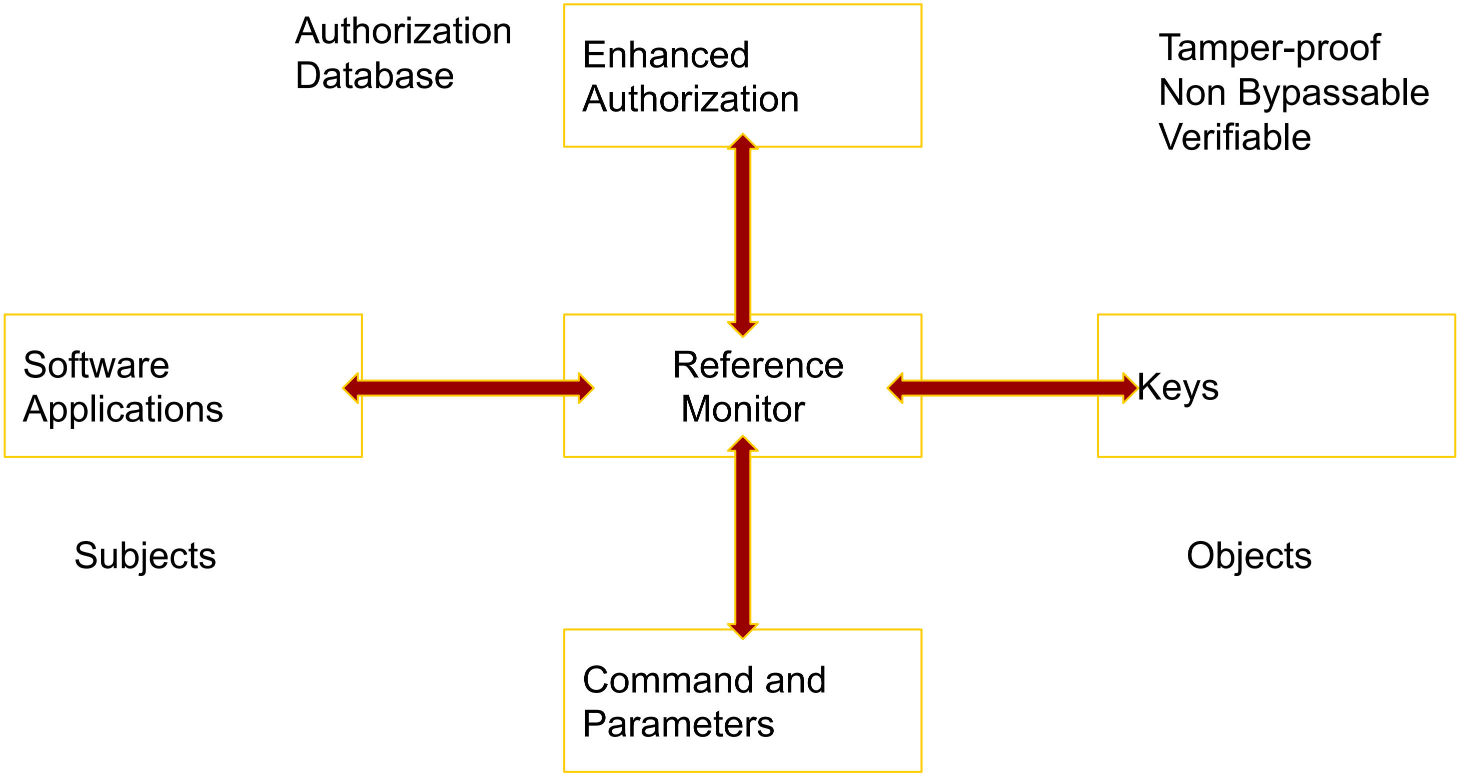
SHE can be easily integrated as compared to HSM.

Acts similar to light EVITA project in HSM (3 levels: light, medium, full).

Both SHE and HSM access the interface and the supported functionality is dependant on the implementation of the hardware used.

Potential Threats

- Missing authorization during runtime
- Pre manufacturing defects
- Post manufacturing defects
- External threats



Authorization
Database

Enhanced
Authorization

Tamper-proof
Non Bypassable
Verifiable

Software
Applications

Reference
Monitor

Keys

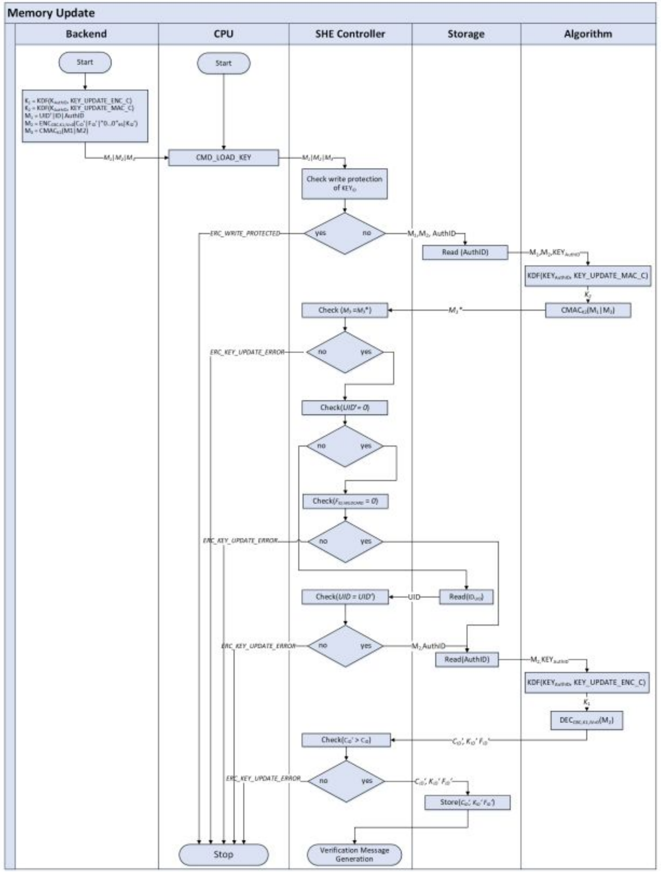
Subjects

Objects

Command and
Parameters

Audit trail

Flow process for memory update protocol


Memory Update
Backend
CPU
SHE Controller
Storage
Algorithm
Start
K_1 = KDF(K_{AuthID}, KEY_UPDATE_ENC_C)
K_2 = KDF(K_{AuthID}, KEY_UPDATE_MAC_C)
M_1 = UID'|ID|AuthID
M_2 = $ENC_{CBC,K1,IV=0}$(C_{ID}'|F_{ID}'|"0...0"$_{95}$|K_{ID}')
M_3 = $CMAC_{K2}$(M1|M2)
$M_1|M_2|M_3$
Start
CMD_LOAD_KEY
$M_1|M_2|M_3$
Check write protection of KEY_{ID}
ERC_WRITE_PROTECTED
yes
no
M_1,M_2, AuthID
Read (AuthID)
M_1,M_2,KEY_{AuthID}
KDF(KEY_{AuthID}, KEY_UPDATE_MAC_C)
K_2
CMAC$_{K2}$(M_1|M_2)
M_3*
Check (M_3 =M_3*)
ERC_KEY_UPDATE_ERROR
no
yes
Check(UID' = 0)
no
yes
Check($F_{ID,WILDCARD}$ = 0)
ERC_KEY_UPDATE_ERROR
no
yes
Check(UID = UID')
UID
Read(ID_{UID})
ERC_KEY_UPDATE_ERROR
no
yes
M_2,AuthID
Read(AuthID)
M_2,KEY_{AuthID}
KDF(KEY_{AuthID}, KEY_UPDATE_ENC_C)
K_1
$DEC_{CBC,K1,IV=0}$(M_2)
Check(C_{ID}' > C_{ID})
C_{ID}', K_{ID}' F_{ID}'
ERC_KEY_UPDATE_ERROR
no
yes
C_{ID}', K_{ID}' F_{ID}'
Store(C_{ID}', K_{ID}' F_{ID}')
Stop
Verification Message Generation

Advantages

- Acts as layer of protection in the authenticated environment
- Flexible and affordable
- Easy integration

Limitations

- Operation might fail due to interruption , e.g. power loss
- Asymmetric algorithms are not supported
- Concurrency is not supported

References

https://www.autosar.org/fileadmin/user_upload/standards/foundation/19-11/AUTOSAR_TR_SecureHardwareExtensions.pdf

https://tremend.com/blog/engineering-insights/how-to-easily-integrate-authentication-and-encryption-using-she-and-hsm/

https://support.vector.com/kb?id=kb_article_view&sysparm_article=KB0012486&sys_kb_id=80fb97b9879c49108816dd383cbb3548&spa=1

https://www.itwissen.info/en/secure-hardware-extension-SHE-126862.html#gsc.tab=0

https://mulloverthing.com/what-is-the-difference-between-she-and-hsm/

Thank You