SITARA™ ARM® PROCESSORS

Boot camp

TEXAS INSTRUMENTS

Linux Cryptography overview and How-to's using OpenSSL

In this session, we will cover cryptography basics and explore cryptographic functions, performance and examples using OpenSSL.

LAB: http://processors.wiki.ti.com/index.php/Sitara_Linux_Training:_Cryptography

July 2012

Creative Commons Attribution-ShareAlike 3.0 (CC BY-SA 3.0)

creative commons

You are free:

to Share – to copy, distribute and transmit the work

to Remix – to adapt the work

to make commercial use of the work

Under the following conditions:

Attribution – You must give the original author(s) credit

Share Alike - If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

With the understanding that:

Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

CC BY-SA 3.0 License: http://creativecommons.org/licenses/by-sa/3.0/us/legalcode

CC BY-SA

Agenda

- Cryptography Is/Is NOT
- Cryptography 101
- Crypto Software Stack
- Open Source Projects
  - OpenSSL
  - OpenSSH
  - OpenSwan
- Cryptographic Hardware Acceleration
- Example Applications

Pre-work check list

- ❑ Installed and configured VMWare Player v4 or later

- ❑ Installed Ubuntu 10.04

- ❑ Installed the latest Sitara Linux SDK and CCSv5

- ❑ Within the Sitara Linux SDK, ran the setup.sh (to install required host packages)

- ❑ Using a Sitara EVM, followed the QSG to connect ethernet, serial cables, SD card and 5V power

- ❑ Booted the EVM and noticed the Matrix GUI application launcher on the LCD

- ❑ Pulled the ipaddr of your EVM and ran remote Matrix using a web browser

- ❑ Brought the USB to Serial cable you confirmed on your setup (preferable)

IS / IS NOT

Is

- All Sitara devices
- Supported in all Sitara SDKs with Opens Source SW.
- AES, DES, 3DES*, SHA1, SHA2, MD5, RNG* hardware accelerators in some GP devices
  - AM35x
  - AM37x
  - AM335x*
- Support for OpenSSL, OpenSSH, Openswan (IPSec)

Is Not

- High Security (HS) silicon support
- Run-time Security

Cryptography 101

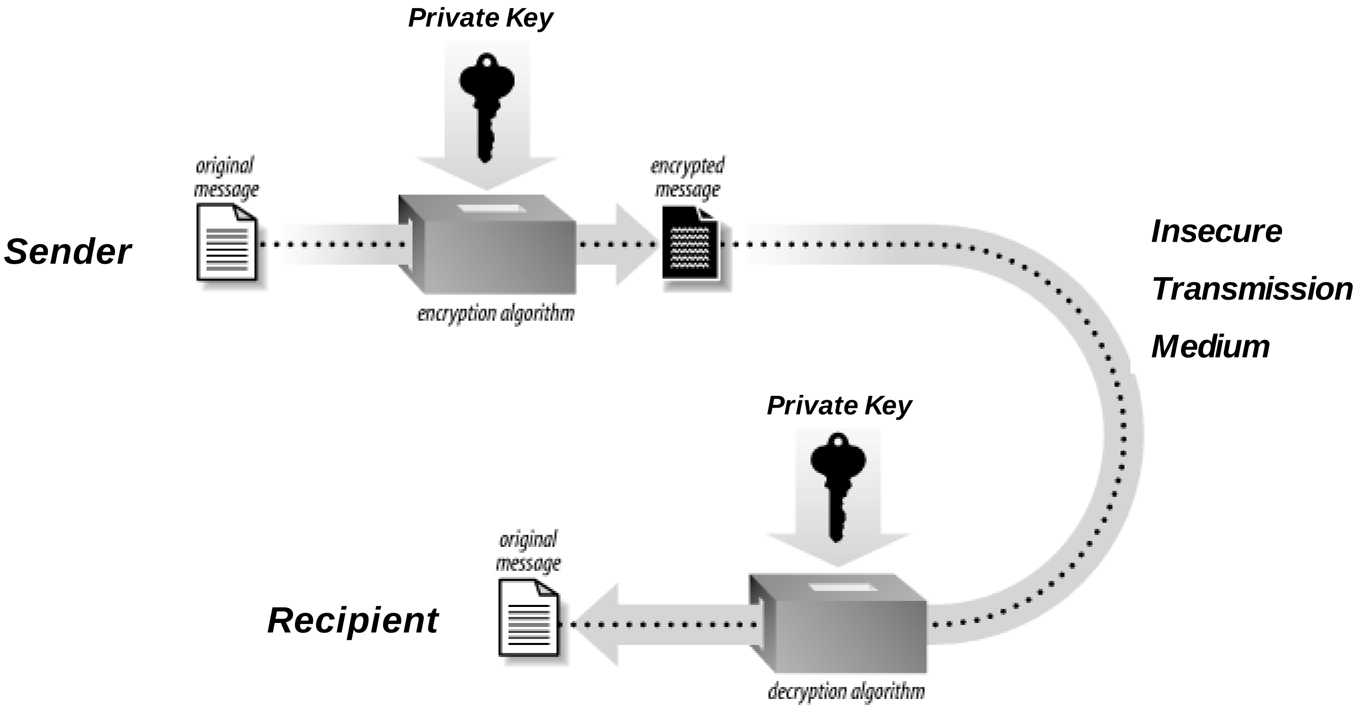
- Definition – Practice and Study of Hiding Information (from Wikipedia)
  - http://en.wikipedia.org/wiki/Cryptography
- Goals
  - Confidentiality
  - Data Integrity
  - Authentication
  - Non-repudiation
- Classic Cryptography
  - Code Book
- Modern Cryptography
  - Public Algorithms
    - **Encrypt/Decrypt - DES, 3DES, AES**
    - **Hash - SHA/MD5**
  - Key and Certificate Generation, Signing, Authentication

Goal #1 – Confidentiality

- Keep the meaning of a message private from unintended viewers in a communication channel
- Accomplished with the use of Key Ciphers (symmetric or asymmetric)
- Intended receiver does not know if the message is complete or altered
- Using a Cipher on clear text produces cipher-text

Modern Cryptography Topics

- Symmetric Key Cryptography

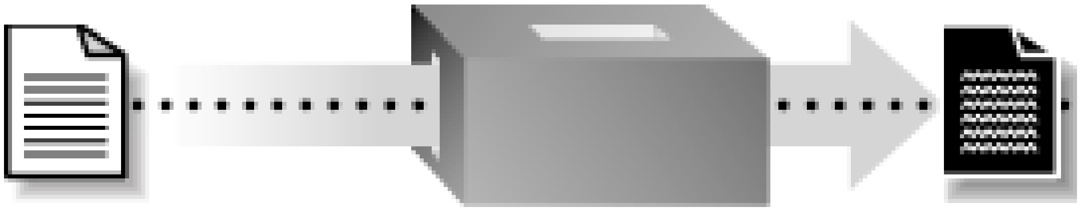


Goal #2 – Data Integrity

- Ensure that a message has not been altered or truncated during transmission
- Only information channel errors considered, no active malicious participants
- One-way Hash functions used to provide integrity
- The output of a Hash function is a fixed length message digest

Modern Cryptography Topics

- Hash function



Goal #3 - Authentication

- Ensure that a message has not been altered or truncated during transmission (same as Data Integrity goal)

- However, now it is assumed there are active malicious elements trying to subvert the message

- Use of Message Authentication Functions (MAC) as a Digital Signature

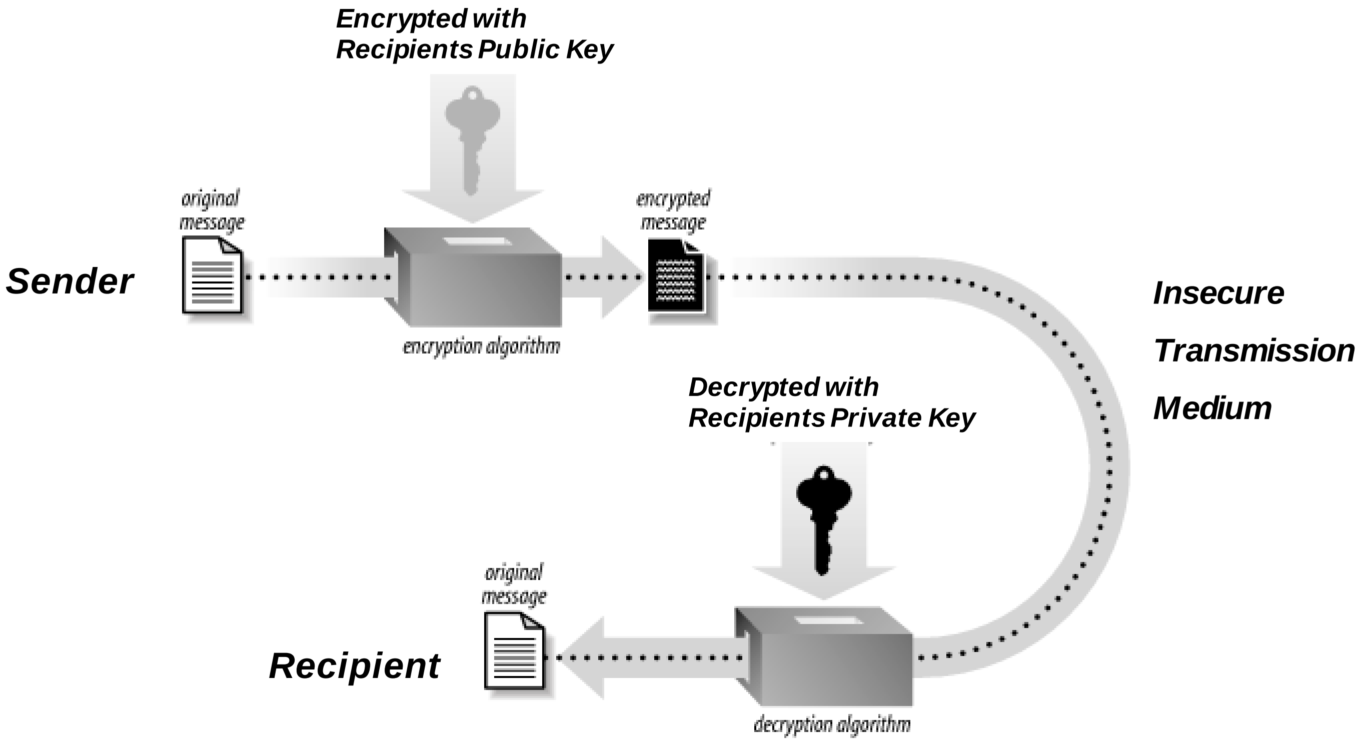
- The output of a MAC is a message tag

Goal #4 – Non-repudiation

- Providing a binding transaction
- Prevent any party involved in a transaction to refute that they took part in the transaction.
- Public Key Digital Signatures
- Asymmetric Public Key Algorithms
- The output of a signature algorithm is a signature

Modern Cryptography Topics

- Asymmetric Key Cryptography - Encryption



Encrypted with Recipients Public Key

Sender

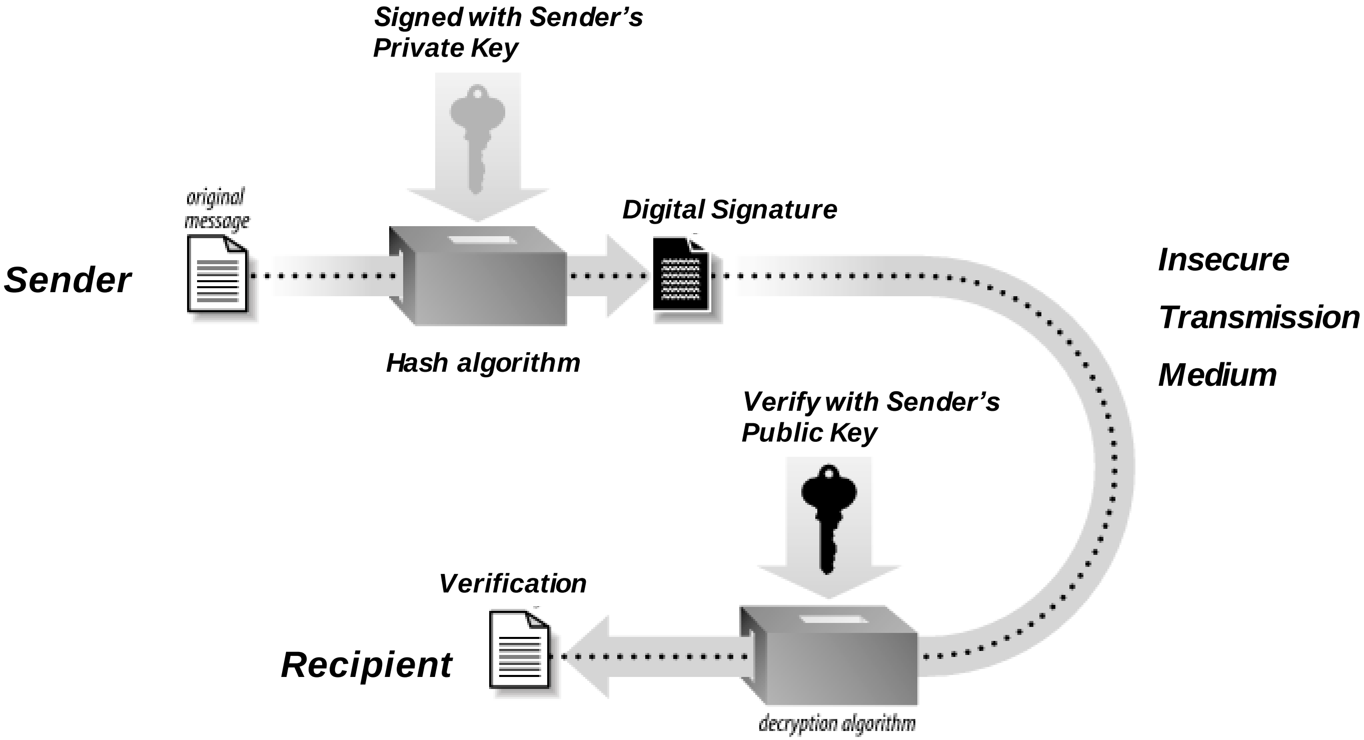
Insecure Transmission Medium

Decrypted with Recipients Private Key

Recipient

Modern Cryptography Topics

- Asymmetric Key Cryptography – Authentication/Signing



Modern Cryptography Topics

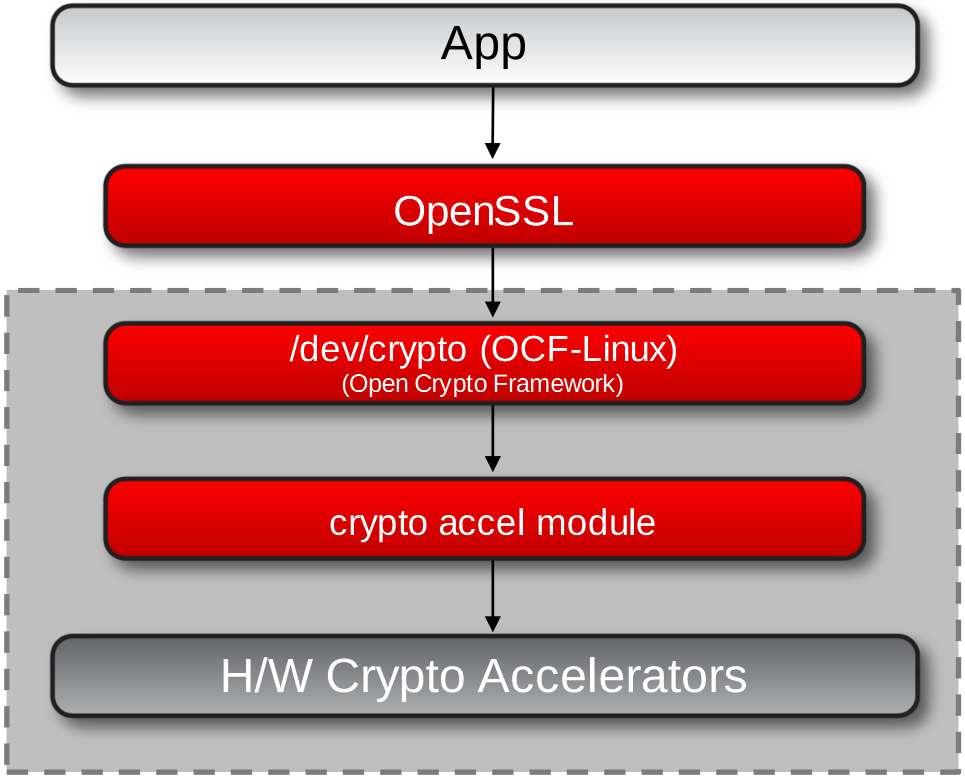
- Encryption/Decryption
  - AES - http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
  - DES/3DES - http://en.wikipedia.org/wiki/Data_Encryption_Standard

- Cryptographic Hash Functions
  - Hash (also called Digest, Fingerprint or Checksum)
  - SHA - http://en.wikipedia.org/wiki/Secure_Hash_Algorithm
  - MD5 - http://en.wikipedia.org/wiki/MD5

- Message Authentication Codes
  - Keyed Hash
  - HMAC – supported by OpenSSL

- Digital Signatures
  - Use of sender's private key to encrypt.
  - DSA – supported by OpenSSL

OpenSSL Crypto SW Stack

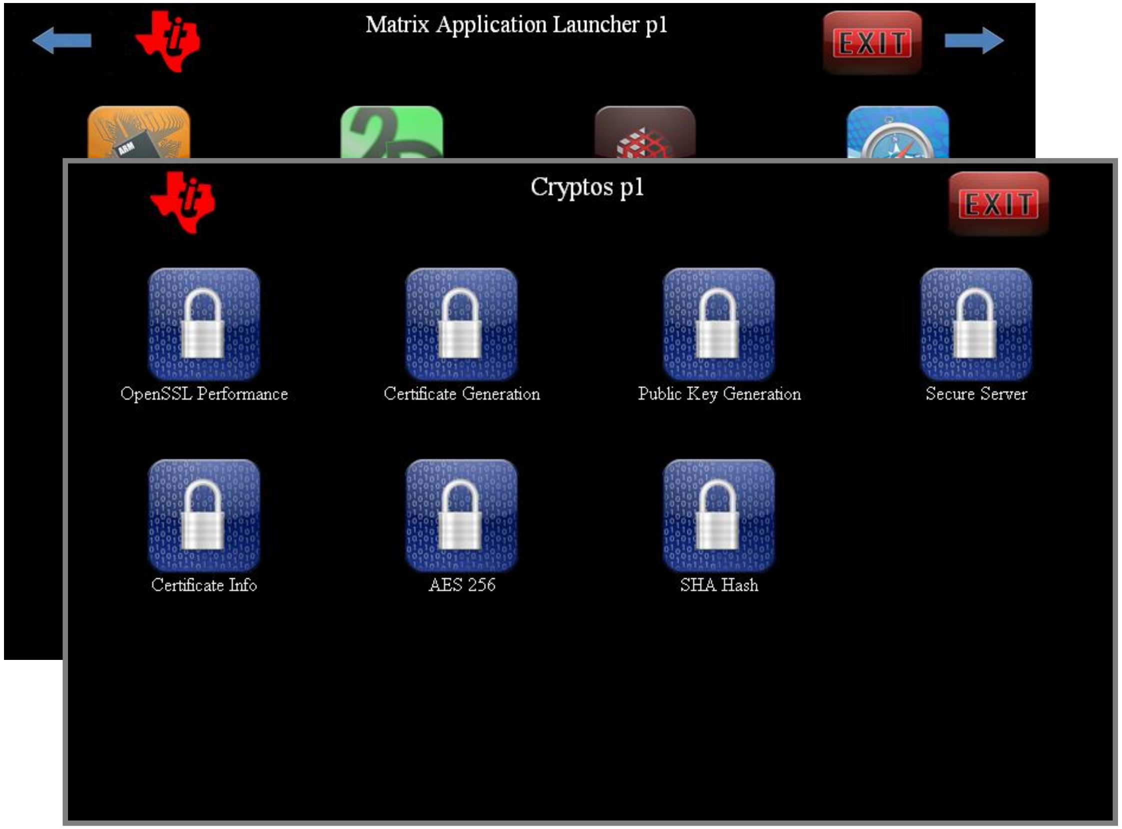


🟥 TI/Open Source SW

⬜ Customer SW

- Open SSL
  - Standard API interface
  - Implements crypto functions in SW
  - Can use OCF when HW is available
- OCF Driver (Open Source module)
  - /dev/crypto created by OCF module
  - Abstracts an API to higher level apps (OpenSSL)
- Crypto accel module (TI)
  - Low level device driver
- TI H/W Crypto Accelerators *
  - AES
  - DES/3DES
  - SHA1/MD5
  - RNG

Cryptography



Example Apps

- OpenSSL
  - Command line application
  - Crypto library can be called from C applications
- Applications
  - Performance
  - Basic Encrypt/Decrypt
  - Basic Hash
  - Private Key/Certificate Generation
  - Public Key Generation
  - Extract/Verify certificate info
  - Connect to Secure Server containing generated certificate

Crypto User's Guide

- ARMCRYPTO (search part number "ARMCRYPTO" at www.ti.com)
  - (http://www.ti.com/tool/armcrypto)

- OpenSSL v1.0.0d
  - Open Source project (http://www.openssl.org/)

- OCF-Linux
  - Open Source project **(http://ocf-linux.sourceforge.net/)**

- ARM Crypto module (omap3_crypto module)
  - TI internally developed
  - External GForge project
  - https://gforge.ti.com/gf/

- Documentation
  - http://processors.wiki.ti.com/index.php/Cryptography_Users_Guide
  - http://processors.wiki.ti.com/index.php/Build_OpenSSL_for_Sitara
  - http://processors.wiki.ti.com/index.php/Build_OCF_for_Sitara
  - http://processors.wiki.ti.com/index.php/Build_Crypto_Module_for_Sitara

Crypto User's Guide

- Command Line Interface to OpenSSL
  - http://www.madboa.com/geek/openssl/

- OpenSSL API for Applications in C
  - http://www.openssl.org/docs/crypto/crypto.html

- OpenSSL in other languages
  - http://www.opensslbook.com/
  - Java
  - Perl
  - Python

LAB

Lab – Cryptography

- In this lab exercise you will run the OpenSSL binary from a command line to…

1. Execute speed tests to analyze performance
2. Perform Basic Encryption/Decryption
3. Perform Basic Hash Functions
4. Generate Asymmetric Key Pair
5. Generate Web Certificate from Key Pair
6. Run Secure Web Server Using Web Certificate

http://processors.wiki.ti.com/index.php/Sitara_Linux_Training:_Cryptography

SITARA™ ARM® PROCESSORS

Boot camp

For more Sitara Boot Camp sessions visit:
www.ti.com/sitarabootcamp

THANK YOU!