

Board ID Tutorial 1 - Security Technology Approaches and Features

Introduction

This tutorial is designed to provide an introductory overview of security technologies and features.

Contents

BOARD ID TUTORIAL 1 - SECURITY TECHNOLOGY APPROACHES AND FEATURES.....	1
INTRODUCTION	1
CONTENTS	1
INTRODUCTION	2
CRYPTOGRAPHIC PROCESSING	2
SYMMETRICAL ENCRYPTION.....	2
ASYMMETRICAL ENCRYPTION (E.G. PUBLIC KEY CRYPTOGRAPHY).....	3
PROTECTION OF DATA	4
LICENSING AND KEY MANAGEMENT.....	5
SECURE SYSTEM DESIGN	8

Introduction

Renesas security chips provide security via the following technologies:

- Cryptographic processing
 - Technology to manage secret information (e.g. encrypting and decrypting data)
 - Keys are used to provide access to data
- Protection of data
 - Protecting keys
 - Protecting sensitive / unsecured data with cryptography
- General purpose CPU
 - Providing functionality equivalent to a microcontroller
 - IO may be limited
- Licensing and key management
 - Creating and Managing (storing, transmitting, receiving) cryptographic keys
 - Creating licenses to use in secure applications
- Secure System Design (integrating Security Chips into a good design)
 - Creating a secure system design which takes advantage of the Renesas Security Chips
 - The Security Chips can be shipped from Renesas with security features to be used in your design, or you can create custom firmware for your system

Cryptographic Processing

Cryptographic processing of data is used for tasks such as:

- Ensuring confidentiality – encrypting and decrypting data
- Ensuring integrity – ensuring data has not been changed
- Ensuring identity – proving that you are who you claim to be
- Ensuring signatures – proving that you signed a document

Some tasks involve all the above, for example: to put your signature on a document, make sure that no unauthorized users be able to read the document, and nobody to change the document signed by the legitimate issuer. Furthermore perhaps the parties receiving the document may also want to make sure the signed document cannot be modified.

There are various types of cryptography used in security chips, but we will briefly illustrate a few here. This will be relevant in our analysis of attacks, and the benefits of various design approaches.

Symmetrical Encryption

- Encryption and decryption of data uses the same key (i.e., Both parties in an exchange of data must have the key). These keys are commonly referred to as “secret keys” and sometimes “shared secrets”
- Keys are extremely sensitive and very special care should be taken to protect the key. It is common for the same key to be used in all installations of an application. In such a case, any failure anywhere will compromise the whole system
- Good performance
- Most common examples in use today: 3DES, AES

Asymmetrical Encryption (e.g. Public Key Cryptography)

- Keys come in pairs, if you encrypt with one key, you decrypt with the other
- One of the keys is secret (referred to as a private key), and one is not secret (referred to as a public key)
- Public keys are not sensitive
- Private keys should be protected in the same way as symmetrical keys (should never be compromised), but the damage due to the loss of a private key is typically not as significant as the loss due to a symmetrical key, because each entity using the key (user or device) has its own private key
- Performance on an embedded system requires a co-processor or special processor
- Most common examples in use today: RSA, ECC (Elliptical Curve Cryptography)

The difference between Asymmetrical and Symmetrical is:

Symmetrical	Asymmetrical
The same key is used for encryption and decryption	Keys come in pairs, if you encrypt with one key you decrypt with the other, and vice versa
All keys are secret (private)	One of the keys is secret (private) and one is available to everybody (public)
Relatively fast, can run on almost any piece of hardware.	Very slow. Runs OK on a Pentium, but not on smaller processors.
Can encrypt large quantities of data	Used with small quantities of data (e.g. 256 bytes or less)
Generally used to encrypt	Generally use to validate

data	identities, ensure data has not changed, and to exchange symmetrical keys between 2 or more parties
------	---

The advantages of Asymmetrical encryption when used in an application are:

- Secret keys do not have to be widely distributed. The party with the least secure environment, for example, would use a public key
- Ideal for establishing identity in a public environment where the parties don't know each other. Public keys are distributed to 3rd parties, where the party distributing the keys keeps the private key
- Used to exchange symmetrical keys. Public Key Cryptography runs slow and only works on small quantities of data. Symmetrical encryption is much faster, but has the limitation that each party has to share the same key. A temporary (session key) can be generated by one party and sent to the other party using the other parties public key. That party then can decrypt the session key with their private key.
- Updates can be done securely offline. With symmetrical key, any secure updates can only be applied with a facility that has the master symmetrical key.

Protection of Data

Encrypted data does not have to be protected, but the keys used to encrypted/decrypt the data have to be protected. It's possible to encrypt keys, but the keys used to encrypt the keys must be protected. Thus, there is always at least one key, usually more, that has to be protected. Data and keys can be protected by encryption utilities such as PGP, and the keys used to protect that data can be remember, as a password. However, such passwords tend to be short in length and not secure enough for best-in-class security. Furthermore, such data can be copied at any time, the number of copies and location of copies cannot be tracked.

Security Chips such as those provided by Renesas, in addition to providing cryptographic operations, also physically protect the data. This provides the following advantages:

- Some data will never be released and thus cannot be copied (e.g. private keys)
- Some data will only be used upon authorization (e.g. from PIN, private keys). If the wrong PIN is entered too many consecutive times, the device may be locked.
- Some data will only be released upon authorization and exchanged with another security device via a key exchange mechanism (e.g. a symmetrical key)
- Some data will be maintained in the processor, then released with a digital signature which asserts that the data is authentic

The security of data can be categorized as:

- Logical
 - The protocol will not release the data over the I/O bus without security
- Electrical/magnetic
 - Techniques to analyze the electrical and magnetic fields will not yield useful information
 - Techniques to analyze micro changes in power consumption or voltage will not yield useful information
- Timing
 - Techniques to analyze micro differences in the timing of operations will not yield useful information
- Physical
 - The device has physical resistance to prevent access to the silicon (drilling, acid, temperature range, etc)
 - The silicon is scrambled to obfuscate any data which might be presented to a hacker if all other countermeasures fail.

Many security processors (secure EEPROM/Flash, secure embedded controllers, cryptographic modules) provide security of data that is limited to logical protection. Certifications such as FIPS140-2 and Common Criteria (along with a rigorous protection profile) are granted to devices with all the above protections for applications that need those protection (e.g. critical sensitive data). Such devices have fundamentally different architectures than other devices, and thus are separate devices and cannot be integrated into other chips. These devices are kept small and inexpensive, and thus tend to have limited CPU speeds and memory. The small incremental cost is justified by the significant increase in security. However, it is important to ensure that the security design is commensurate with the security level of the device.

Licensing and Key Management

Best-in-class security may include a security chip in the design. There are a variety of architectures to address typical security applications, but a common requirement is that security devices have to be loaded with keys for applications such as authentication, digital signatures, encryption, etc. Your solution will most likely require your security chips to be loaded with keys. This will be done by an Identity / License Authority via a third party or via secure tools you run in your operations.

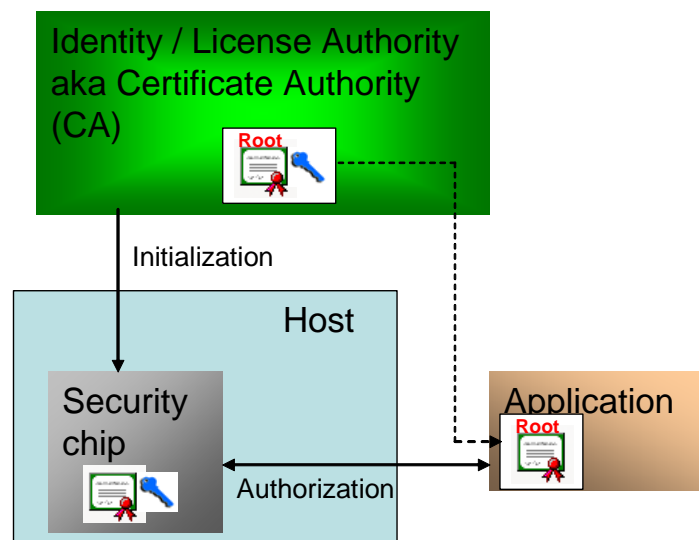
The techniques vary depending on the application and whether you are using symmetrical or Public Key technologies, but the steps are similar. Examples:

Symmetrical Keys

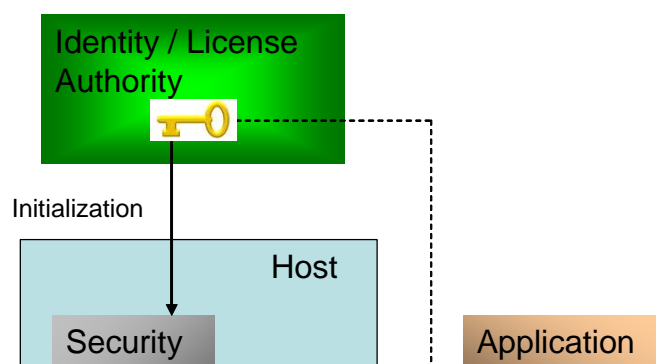
- The Identity / License Authority contains a secret master key
- It generates unique information (e.g. serial number) which is sent to the security chip along with the secret key during Initialization / Manufacturer.

- This should be done in a secure environment (e.g. the communications between the Authority and the Chip should not be compromised)
- There are typically two approaches
 - The chip is initialized in a secure facility then soldered or plugged into the board
 - The chip is initialized while in the board
- The same master key is loaded into the application. This key must be secured. Additionally, in some cases a security chip can be used to protect the key
- Authorization between the application and the security occurs with a challenge / response mechanism to validate the Unique Information in the security chip. (Examples of how this is done is included in the demos)
- Note that the use of a single master key through the entire system is a key vulnerability. If the master key is compromised at any point, the entire system is compromised. This is a *Break Once Fail Everywhere (BOFE)* scenario

Public Key



Public Key, continued



This approach has the same basic steps, but several advantages:

- The same key is NOT used everywhere
- The weakest security for keys is typically the application, and it only uses a public root key. This has to be protected against being changed, not viewed
- The security chip has a certificate which contains one or more piece of unique identifying information.
- The private key proves only ownership of that certificate. This key is never released, so it is not possible for one chip to masquerade as another as they have different keys sets
- The certificate contains the matching public key. The keyset is generated in one of two ways:
 - Generated inside the Security chip, and the public key is sent to the Identity / License Authority for a certificate to be created.
 - Generated inside the Identity / License Authority and “injected” into the Security chip. This is faster in a manufacturing environment because key generation is much faster in the License Authority than the chip
- The certificate contains the following important information
 - Public key
 - Secure Identifying fields (fields that have been validated)
 - Non-secure fields (fields that are for reference only)
 - Digital signature on the above (signed by the private key of the CA which is a critical secret key)
 - Note: Certificates used in our demo are mini-certificates which contain a minimal amount of information for the in order to save processing resources. X.509 standard certificates could also be used, but they require more resources to store and process.
- Security Measures can be taken to ensure the Security Chip is an authentic chip before the keys are loaded

The use of Public Key cryptography in conjunction with technologies such as certificates and Certificate Authorities is collectively referred to as Public Key Infrastructure (PKI).

Secure System Design

Security Chips can significantly enhanced your security by protecting critical assets such as identification, licensing data, system and application keys. However, the design of your total solution will be a more significant factor in the security of your overall design. For example a security chip can do an authentication, but how will this authentication be enforced at the system level? Certifications such as FIPS140-2 and Common Criteria are generally limited to the components of your system. The Common Criteria EAL certification levels give a good example of the kinds of decisions to make in a design.

A good design will generally include the following steps:

- Business Case / Objectives
- Understanding of how business processes affect security
- An analysis of assumptions, risks and decisions as to what threats will be addressed (i.e., determining the appropriate security level)
- Choosing among of variety of design approaches
- Peer review
- Final design

More Resources

<http://america.renesas.com/boardid>



Copyright © Renesas Technology Americas Ltd. All rights reserved