Hybrid Crypto Hardware Utilizing Symmetric-Key & Public-Key Cryptosystems

- Adnan Abdul-Aziz Gutub
  - Umm Al-Qura University
- Farhan Abdul-Aziz Khan
  - King Fahd University of Petroleum & Minerals
Outline

• Introduction
• Motivation
• Crypto system design & modeling
• DES & AES cipher & inverse cipher design
• RSA & Pseudorandom number generator design
• GCD unit design
• Conclusion
Introduction

• Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication and data origin authentication.

• Cryptography is not the only means of providing information security but rather one set of techniques.

• A crypto system is a general term referring to a set of cryptographic primitives used to provide information security services. Most often, the term is used in conjunction with primitives providing encryption and decryption.
Secure Communication

Alice \rightarrow Encrypt \rightarrow Ciphertext \rightarrow Decrypt \rightarrow Bob

Encryption Key

Decryption Key

Plaintext

Mallory
Oscar

Eve

Enemy or Adversary
Classic Cryptography

• Substitution (Caesar)
  • Transposition
• Enigma Machine
  • Vigenere
• Block (Hill)
• Vernam (one time pad)
  • DES
  • AES
Symmetric key algorithms

• Encryption & decryption keys are known to both
• They are usually related (if not identical)
  – easy to derive the decryption key once the encryption key is known
• DES, AES (Rijndael)
• A secret must be known (agreed upon) to communicating parties
  – so they can generate encryption and decryption keys

Key distribution and/or management problem

Sender

Receiver

Key = Z ← Same → Key = Z

Message

Sender

Receiver

Key = Z ← Same → Key = Z

Message

Lock

Z

Z
Symmetric-key cryptography

- encryption/decryption is performed using one secret key: e.g. DES and AES.
- Advantage: they are faster in execution
  - straightforward transformations.
  - can be pipelined to give better performance.
- Problem is to find efficient method to exchange keys securely: key distribution problem.
- If secret key is disclosed during key exchange >>> whole symmetric key crypto algorithm >>> >>> completely vulnerable.
Simple Example of PKC
Non-mathematical

Bob has a box and a padlock which only he can unlock
once it is locked

Alice wants to send a message to Bob

Bob sends its box and the padlock unlocked to Alice

Alice puts its message in the box and locks the box
using Bob's padlock and sends it to Bob

– thinking that the message is safe since it is Bob that can
unlock the padlock and access the contents of the box.

Bob receives the box, unlocks the padlock and reads the
message

OPEN DIRECTORY

<table>
<thead>
<tr>
<th>SENDER</th>
<th>RECEIVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message</td>
<td>Message</td>
</tr>
<tr>
<td>K-close</td>
<td>K-open</td>
</tr>
<tr>
<td>K-close (keep secret)</td>
<td>K-open</td>
</tr>
</tbody>
</table>

Attack: Eve can replace Bob's padlock with hers on the way
Public-key cryptography

• Public-key crypto algorithms (asymmetric cryptographic algorithms) e.g. RSA use separate public and private keys to perform encryption and decryption.
• The public key is made widely available and is used by communicating parties to encrypt data. Only the party with the correct corresponding private key can decrypt data.
• Public-key algorithms are the most secure cryptographic algorithms because they are based on an underlying mathematically hard-to-solve problem like integer factorization problem, discrete logarithm problem etc.
• They are also substantially slower than symmetric-key cryptography algorithms.
• Public key algorithms are commonly used in practice for the transport of keys subsequently used for bulk data encryption and decryption by symmetric-key algorithms and other applications.
Basic Cryptographic Applications

- **Confidentiality** – Hiding contents of messages exchanged in a transaction.
- **Authentication** – Ensuring that the origin of a message is correctly identified.
- **Integrity** – Ensuring that only authorized parties are able to modify computer system assets and transmitted information.
- **Non-repudiation** – Requires that neither of the authorized parties deny the aspects of a valid transaction.
Other Cryptographic Applications

- **Digital Signatures**
  - allows electronically sign (personalize) the electronic documents, messages and transactions

- **Identification**
  - is capable of replacing password-based identification methods with more powerful (secure) techniques

- **Key Establishment**
  - To communicate a key to your correspondent (or perhaps actually mutually generate it with him) whom you have never physically met before

- **Secret Sharing**
  - Distribute the parts of a secret to a group of people who can never exploit it individually
Other Cryptographic Applications

- **E-commerce**
  - carry out the secure transaction over an insecure channel like Internet
- **E-cash**
  - The cash can be sent securely through computer networks
  - The cash cannot be copied and reused
  - The spender of the cash can remain anonymous
  - The transaction can be done offline
  - The cash transferred to others
  - A piece of cash can be divided into smaller amounts
- **Games**
  - Flipping coins over the phone
- **Electronic Voting**
Motivation

- One approach for design of a hybrid crypto system is to combine the strengths of both symmetric and public-key algorithms.

- This can be done by performing data encryption and decryption using symmetric algorithms like DES and AES and key encryption using public-key algorithm like RSA.

- This will make encryption/decryption faster and at the same time will make the key distribution more secure over an insecure channel.
Traditional Crypto system design

Encryption Block

Decryption Block

Key

Plain Text

Cipher Text

Plain Text

Cipher Text

Key
Proposed Crypto system design

Encryption Block

Key Encryption Block

Decryption Block

Key Decryption Block

Plain Text

Cipher Text

Symmetric Key

Encrypted Key

Public Key

Cipher Text

Decrypted Key

Private Key

Encrypted Key

Symmetric Key

Plain Text
Key Transfer

Key Transfer

Insecure Channel

Public Key

Key Encryption Block

Symmetric Key

Private Key

Key Decryption Block
Crypto system design & modeling

Different modules of crypto system

- DES Cipher
- DES Inverse Cipher
- AES Cipher
- AES Inverse Cipher
- RSA Cipher
- GCD Unit
- Pseudorandom Number Generator
Modules of hybrid crypto system

Source

Destination
DES

- Was designed to encipher sensitive but unclassified data

- A block cipher (64 bits):
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext

- A product cipher
  - basic unit is the bit
  - performs both substitution and transposition (permutation) on the bits

- Cipher consists of 16 rounds (iterations) each with a round key generated from the user-supplied key

- Round key = 48 bits
Generation of Round Keys

- Parity bit dropped ⇒ 56 bits
- $PC_1$ & $PC_2$ = permutation tables
- $LSH$ = Left shift (rotations)
- $K_1$,...,$K_{16}$ = Round keys = 48 bits each
DES cipher design

- Simulation of Area optimized DES cipher
DES cipher design

- Simulation of performance optimized DES cipher

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>16384</td>
<td>65535</td>
<td>65535</td>
<td>65535</td>
<td>65535</td>
<td>65535</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>8232ba6beab6692</td>
<td>8000000000000000</td>
<td>4000000000000000</td>
<td>20000000000000000</td>
<td>100000000000000000</td>
<td>0000000000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17</td>
<td>17</td>
</tr>
</tbody>
</table>
Encipherment

- **Input** = 64 bits
- **Output of rounds 1 = input of round 2**
- **Round input is partitioned into L & R = 32 bits each**
- **R is to be extended to 48 bits.**
- **f runs on R & K = 48 bits producing 32 bits output to be XOR’d with L**
- **XORing Output (32 bits) ⇒ expanded to 48 bits ⇒ new R**
- **Previous R ⇒ new L**

**Strength of DES is in Function f**
DES inverse cipher design

- Simulation of Area optimized DES inverse cipher
DES inverse cipher design

- Simulation of performance optimized DES inverse cipher
Rijndael Overview

- Block size is also variable (128/192/256)
- # of rounds is a function of key length:

<table>
<thead>
<tr>
<th>Key length (in bits)</th>
<th>#of rounds $n_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>10</td>
</tr>
<tr>
<td>192</td>
<td>12</td>
</tr>
<tr>
<td>256</td>
<td>14</td>
</tr>
</tbody>
</table>
Rijndael AES Crypto operation

Encryption

Decryption

Round $n_r$

Key Addition Layer

Byte Substitution Layer

ShiftRow Layer

MixColumn Layer

Key Addition Layer

Diffusion layer

Rounds $1 \ldots n_r - 1$

Key Addition Layer

InvShiftRow Layer

InvByteSub Layer

Inverse of rounds $1 \ldots n_r - 1$

Key Addition Layer

InvShiftRow Layer

InvMixColumn Layer

InvByteSub Layer

Round $n_r$

Byte Substitution Layer

ShiftRow Layer

Key Addition Layer
AES cipher design

- Simulation of AES Cipher
AES inverse cipher design

• Simulation of AES inverse cipher
Overview of Public Key Cryptosystem

- Integer factorization problems (RSA)
  - factoring large integers
- Discrete Logarithm problems (Diffie-Hellman, ElGamal)
- Elliptic Curve Cryptosystems

<table>
<thead>
<tr>
<th>Algorithm family</th>
<th>Bit length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer Factorization (IF)</td>
<td>1024</td>
</tr>
<tr>
<td>Discrete Logarithm (DL)</td>
<td>1024</td>
</tr>
<tr>
<td>Elliptic curves (EC)</td>
<td>160</td>
</tr>
<tr>
<td>Block cipher</td>
<td>80</td>
</tr>
</tbody>
</table>
RSA

• 1978 @ MIT: Rivest Shamir Adleman = RSA
  ▫ 2 years after Diffie Helman idea was proposed
• Exponentiation cipher
• Based on *Integer Factorization* problem
  ▫ Relies on the difficulty of determining the number of numbers relatively prime to a large integer $n$
RSA Algorithm

- **Choose**: \( p, q \in \text{positive distinct large primes} \)
- **Compute**: \( n = p \times q \)
- \( n \) = encryption/decryption modulus \( \rightarrow \) computations in \( Z_n \)
- **Compute**: \( \phi(n) = (p - 1)(q - 1) \)
- **Choose randomly**: \( e \in Z_{\phi(n)}^* \)
  \( \rightarrow \gcd(\phi(n), e) = 1 \), \((e \text{ has an inverse mod } \phi(n))\)
- Find \( d = e^{-1} = ?? \mod \phi(n) \)
  - Compute \( d \) such that \( ed \mod \phi(n) = 1 \)

- **Encryption**: \( c = x^e \mod n \) where \( x < n \)
- **Decryption**: \( x = c^d \mod n \)
- \( n, e \) are made public but \( p, q, d \) are secret
RSA encryption

- Encryption with $N=3233$, $m=123$, $e=17$
RSA decryption

- Decryption with $N=3233$, $c=855$, $d=2753$
Greatest Common Divisor unit

- Simulation of GCD unit
Pseudo-random number generator

- Simulation of pseudo-random number generator
Simulation timing Summary

Simulation is performed using Model Sim SE 5.7e. The total clock cycles taken for different modules is shown in the table.

<table>
<thead>
<tr>
<th>MODULE</th>
<th>CLOCK CYCLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area-optimized DES</td>
<td>16</td>
</tr>
<tr>
<td>Performance-optimized DES</td>
<td>1</td>
</tr>
<tr>
<td>AES Cipher</td>
<td>12</td>
</tr>
<tr>
<td>AES Inverse Cipher</td>
<td>24</td>
</tr>
<tr>
<td>RSA Modular Multiplier</td>
<td>Variable</td>
</tr>
<tr>
<td>GCD unit</td>
<td>variable</td>
</tr>
<tr>
<td>Pseudo-random number generator</td>
<td>4</td>
</tr>
</tbody>
</table>
Synthesis Summary

Synthesis is performed using Synopsys Design Compiler. The total cell area of different modules is shown in the table.

<table>
<thead>
<tr>
<th>MODULE</th>
<th>TOTAL CELL AREA (No. of cells)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area-optimized DES</td>
<td>3132</td>
</tr>
<tr>
<td>Performance-optimized DES</td>
<td>38033</td>
</tr>
<tr>
<td>AES Cipher</td>
<td>28286</td>
</tr>
<tr>
<td>AES Inverse Cipher</td>
<td>52182</td>
</tr>
<tr>
<td>RSA Modular Multiplier</td>
<td>3691</td>
</tr>
<tr>
<td>GCD unit</td>
<td>1285</td>
</tr>
<tr>
<td>Pseudo-random number generator</td>
<td>1334</td>
</tr>
</tbody>
</table>
## Performance & Area of Implemented Versions of the hybrid crypto algorithms

<table>
<thead>
<tr>
<th>Version</th>
<th>Performance (No. of Clock Cycles)</th>
<th>Area (No. of Cells)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area-Optimized DES</td>
<td>16</td>
<td>3132</td>
</tr>
<tr>
<td>Performance-Optimized DES</td>
<td>1</td>
<td>38033</td>
</tr>
<tr>
<td>AES Cipher</td>
<td>12</td>
<td>28286</td>
</tr>
<tr>
<td>AES Inverse Cipher</td>
<td>24</td>
<td>52182</td>
</tr>
<tr>
<td>RSA Modular Multiplier</td>
<td>Variable</td>
<td>3691</td>
</tr>
</tbody>
</table>
Conclusion

- Design of a crypto system is presented utilizing both symmetric-key and public-key cryptographic algorithms.
- The symmetric key algorithms DES and AES are used for data encryption.
- The public key algorithm RSA is used for encrypting the secret key before performing key distribution.
- DES cipher, DES inverse cipher, AES cipher and AES inverse cipher modules are part of the system.
- The crypto system also contains pseudo-random number generator for random key generation and a GCD computation unit for use in RSA cipher.
- Area-optimized and performance-optimized versions for DES and AES algorithms are designed by Verilog HDL.
- Comparison is also presented for making a design choice.
köszönöm  המודה! děkuji mahalo 고맙습니다
thank you  merci 谢谢 danke
شكرا どうもありがとうございます gracias