How to Share Your Secrets

Priscilla Zhu and Garima Rastogi

MIT PRIMES Computer Science Reading Group

Dec 6th, 2022
Overview

1. Secure Communication
   - Terminology
   - Defining Correctness
   - Defining Security

2. Encryption Schemes
   - One-Time Pad
   - Perfect Secrecy

3. Secret Sharing
   - Terminology
   - Correctness and Security
   - Algorithms
   - Example
Eavesdropping Erika

On the planet Osgiliath in a galaxy far, far, away...

Akaali  Message  Blathereen

Erika
Secure Communication
Secret-key Encryption

Components:
- Secret key, \( k \)
- Message \( m \)
- Ciphertext \( c \)
- Key Generation: \( k \leftarrow \text{Gen}(1^n) \)
- Encryption: \( c \leftarrow \text{Enc}(k, m) \)
- Decryption: \( m \leftarrow \text{Dec}(k, c) \)
Purpose

- Secret key $k$ from key space $\mathcal{K}$: $k \leftarrow \mathcal{K}$
- Message $m$ from message space $\mathcal{M}$: $m \leftarrow \mathcal{M}$
- Ciphertext $c$ from ciphertext space $\mathcal{C}$: $c \leftarrow \mathcal{C}$

Algorithms within a cryptographic scheme:
- Key Generation Algorithm: $Gen(1^n)$: $k \leftarrow Gen$
- Encryption Algorithm: $Enc(k, m)$: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$
- Decryption Algorithm: $Dec(k, c)$: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

Purpose: If Akaali sends over $m$ as $c$, Blathereen should be able to use $k$ to correctly determine $m$. 
## Definition of Correctness

**Definition**

An encryption scheme is said to be correct if, for all $k \leftarrow \mathcal{K}$ and $m \leftarrow \mathcal{M}$, $\text{Dec}(k, c = \text{Enc}(k, m)) = m$. 
Definition of Correctness

**Definition**
An encryption scheme is said to be correct if, for all $k \leftarrow \mathcal{K}$ and $m \leftarrow \mathcal{M}$, $Dec(k, c = Enc(k, m)) = m$.

**Non-Example**
Consider $Enc(k, m) = m^k$ and $Dec(k, c) = k\sqrt{c}$. Let $k = 3$. Then, $Dec(k, Enc(k, m)) = 3\sqrt{m^3} = m$. Let $k = 2$. Then, for $m < 0$, $Dec(k, Enc(k, m)) = 2\sqrt{m^2} = -m$. 

Priscilla Zhu and Garima Rastogi
Definition of Correctness

**Definition**
An encryption scheme is said to be correct if, for all $k \leftarrow \mathcal{K}$ and $m \leftarrow \mathcal{M}$, $\text{Dec}(k, c = \text{Enc}(k, m)) = m$.

**Non-Example**
Consider $\text{Enc}(k, m) = m^k$ and $\text{Dec}(k, c) = \sqrt[k]{c}$.

- Let $k = 3$. Then, $\text{Dec}(k, \text{Enc}(k, m)) = \sqrt[3]{m^3} = m$.
- Let $k = 2$. Then, for $m < 0$, $\text{Dec}(k, \text{Enc}(k, m)) = \sqrt{m^2} = -m$. 
Secure Communication

Defining Security

Definition of Security

Shannon’s Perfect Secrecy

\[ \forall \mathcal{M} \; \forall m \in \text{Supp}(\mathcal{M}), \; \forall c \in \text{Supp}(\mathcal{C}), \]

\[ Pr[\mathcal{M} = m | \text{Enc}(K, \mathcal{M}) = c] = Pr[\mathcal{M} = m] \]
Definition of Security

Shannon’s Perfect Secrecy

\[ \forall \mathcal{M} \ \forall m \in \text{Supp}(\mathcal{M}), \ \forall c \in \text{Supp}(\mathcal{C}), \]
\[ \Pr[\mathcal{M} = m | \text{Enc}(K, \mathcal{M}) = c] = \Pr[\mathcal{M} = m] \]

Perfect Indistinguishability

\[ \forall \mathcal{M} \ \forall m, m' \in \text{Supp}(\mathcal{M}), \]
\[ \Pr[\text{Enc}(K, m) = c] = \Pr[\text{Enc}(K, m') = c] \]
Definition of Security

Shannon’s Perfect Secrecy

\[ \forall \mathcal{M} \, \forall m \in \text{Supp}(\mathcal{M}), \forall c \in \text{Supp}(\mathcal{C}), \]
\[ Pr[\mathcal{M} = m|\text{Enc}(K, \mathcal{M}) = c] = Pr[\mathcal{M} = m] \]

Perfect Indistinguishability

\[ \forall \mathcal{M} \, \forall m, m' \in \text{Supp}(\mathcal{M}), \]
\[ Pr[\text{Enc}(K, m) = c] = Pr[\text{Enc}(K, m') = c] \]

Theorem

An encryption scheme \((Gen, Enc, Dec)\) satisfies perfect secrecy if and only if it satisfies perfect indistinguishability.
Encryption Schemes
One-Time Pad

Construction:

**Encryption Schemes** | **One-Time Pad**
--- | ---

One-Time Pad

**Construction:**

Gen:

$k_r \leftarrow \{0, 1\}^n$, thus $|K| = 2^n$

$n$-bit message $m$, thus $|M| = 2^n$

$Enc(k, m)$:

$c = m \oplus k$

XOR bitwise operator: $11 \oplus 10 = 01$ (commutative)

$Dec(k, c)$:

$m = c \oplus k$

$m = c \oplus k = m \oplus k \oplus k = m \oplus 0^n = m$. 
Construction:

- **Gen**: $k \leftarrow \{0, 1\}^n$, thus $|\mathcal{K}| = 2^n$
One-Time Pad

**Construction:**

- **Gen:** $k \leftarrow \{0, 1\}^n$, thus $|\mathcal{K}| = 2^n$
- $n$-bit message $m$, thus $|\mathcal{M}| = 2^n$
One-Time Pad

Construction:

- $Gen: \ k \leftarrow \{0, 1\}^n$, thus $|K| = 2^n$
- $n$-bit message $m$, thus $|M| = 2^n$
- $Enc(k, m): \ c = m \oplus k$
  - XOR bitwise operator: $11 \oplus 10 = 01$ (commutative)
Construction:

- **Gen:** \( k \leftarrow \{0, 1\}^n \), thus \(|\mathcal{K}| = 2^n\)
- \( n \)-bit message \( m \), thus \(|\mathcal{M}| = 2^n\)
- \( Enc(k, m) \): \( c = m \oplus k \)
  - XOR bitwise operator: \( 11 \oplus 10 = 01 \) (commutative)
- \( Dec(k, c) \): \( m = c \oplus k \)

\[
\begin{align*}
m &= c \oplus k \\
   &= m \oplus k \oplus k \\
   &= m \oplus 0^n \\
   &= m.
\end{align*}
\]
One-Time Pad

Perfect Indistinguishability Example

Consider $c = m \oplus k = 1001101$. What is $m$? What is $k$?
Perfect Indistinguishability Example

Consider \( c = m \oplus k = 1001101 \). What is \( m \)? What is \( k \)?

- First digits of \((m, k)\) either \((1, 0)\) or \((0, 1)\)
- Second digits either \((0, 0)\) or \((1, 1)\)
  
  :
One-Time Pad

Perfect Indistinguishability Example

Consider \( c = m \oplus k = 1001101 \). What is \( m \)? What is \( k \)?

- First digits of \((m, k)\) either (1, 0) or (0, 1)
- Second digits either (0, 0) or (1, 1)

Thus, there are \( 2^n \) possibilities for \((m, k)\).
Consider distinct messages $m_1$ and $m_2$.

Then, for the chosen key $k$, their ciphers are:

$c_1 = m_1 \oplus k$ and $c_2 = m_2 \oplus k$.

Information leak:

$c_1 \oplus c_2 = (m_1 \oplus k) \oplus (m_2 \oplus k) = m_1 \oplus m_2 \oplus k \oplus k = m_1 \oplus m_2$.
Two-Time Pad Attack

Consider distinct messages $m_1$ and $m_2$. 
One-Time Pad

Two-Time Pad Attack

Consider distinct messages $m_1$ and $m_2$. Then, for the chosen key $k$, their ciphers are $c_1 = m_1 \oplus k$ and $c_2 = m_2 \oplus k$. 
Two-Time Pad Attack

Consider distinct messages $m_1$ and $m_2$. Then, for the chosen key $k$, their ciphers are $c_1 = m_1 \oplus k$ and $c_2 = m_2 \oplus k$. Information leak:

\[
c_1 \oplus c_2 = (m_1 \oplus k) \oplus (m_2 \oplus k)
= m_1 \oplus m_2 \oplus k \oplus k
= m_1 \oplus m_2.
\]
Perfect Secrecy

**Theorem**

Shannon’s theorem of perfect secrecy: for any perfectly secure scheme, $|\mathcal{K}| \geq |\mathcal{M}|$. 
Perfect Secrecy

**Theorem**

Shannon’s theorem of perfect secrecy: for any perfectly secure scheme, $|K| \geq |M|$.

**Proof:**

Figure 1: Prof. Vinod Vaikuntanathan’s slides for 6.875 at MIT

- Every key is distinct
- One-time pad: $n$-bit message $m$; $k \leftarrow \{0, 1\}^n$
Pseudorandom Generators (PRG)

Pseudorandom Generators: seed $\rightarrow b_1, b_2, b_3...$
Pseudorandom Generators (PRG)

Pseudorandom Generators: seed $\rightarrow b_1, b_2, b_3...$

Definition

A deterministic polynomial-time computable function $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is a PRG if:

1) $m > n$, and
Pseudorandom Generators (PRG)

Pseudorandom Generators: seed → $b_1, b_2, b_3...$

Definition

A deterministic polynomial-time computable function $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is a PRG if:

1) $m > n$, and
2) For every probabilistic polynomial time (PPT) algorithm $D$, there is a negligible function $\mu$ such that:

$$|\Pr[D(G(U_n)) = 0] - \Pr[D(U_m)] = 0| = \mu(n)$$
However...

How can Akaali and Blathereen make sure that the secret stays hidden?
Secret Sharing
Definition

Goal: divide a secret into $n$ components, where at least $1 \leq t \leq n$ components are needed to reconstruct the full secret.
Definition

Goal: divide a secret into $n$ components, where at least $1 \leq t \leq n$ components are needed to reconstruct the full secret.

Definition

An $(n, t)$ sharing scheme consists of:

- **Share(secret $s$):** outputs $\{s_1, s_2, \ldots, s_n\}$
- **Reconstruct($l, \{s_i\}_{i \in l}$):** outputs $s$ if $l \subseteq \{1, 2, \ldots, n\}$ where $|l| \geq t$. 
Correctness

For all secrets $s$,
- Share$(s) \rightarrow \{s_1, s_2, \ldots, s_n\}$
- For any $I \subseteq \{1, 2, \ldots, n\}$ where $|I| \geq t$,
  Reconstruct$(I, \{s_i\}_{i \in I}) \rightarrow s$. 

Security

For all $I \in \{1, 2, \ldots, n\}$ where $|I| < t$,
- $\{s_i\}_{i \in I}$ should reveal no information about $s$. 

Priscilla Zhu and Garima Rastogi
Notions

Correctness
For all secrets $s$,
- $\text{Share}(s) \rightarrow \{s_1, s_2, \ldots, s_n\}$
- For any $I \subseteq \{1, 2, \ldots, n\}$ where $|I| \geq t$,
  $\text{Reconstruct}(I, \{s_i\}_{i \in I}) \rightarrow s$.

Security
For all $I \in \{1, 2, \ldots, n\}$ where $|I| < t$, $\{s_i\}_{i \in I}$ should reveal no information about $s$. 
Two Common Types

Polynomial Construction

Share(s):
n points on the polynomial

Based in Lagrange's interpolation theorem

Theorem

Given k distinct points on a polynomial, we can determine a polynomial of degree d ≤ k − 1.

I.e., t = k

Constant term necessary to reconstruct the secret

Shamir's Secret Sharing Algorithm

Additive Construction

Share(s):
n numbers adding up to an encoding of s

Requires all n people to come together
Two Common Types

Polynomial Construction

- Share($s$): $n$ points on the polynomial

Based in Lagrange's interpolation theorem,

Given $k$ distinct points on a polynomial, we can determine a polynomial of degree $d$ where $d \leq k - 1$.

I.e., $t = k$

Constant term necessary to reconstruct the secret

Shamir's Secret Sharing Algorithm

Additive Construction

Share($s$): $n$ numbers adding up to an encoding of $s$

Requires all $n$ people to come together
Two Common Types

Polynomial Construction

- Share(s): \( n \) points on the polynomial
- Based in Lagrange’s interpolation theorem

Theorem

Given \( k \) distinct points on a polynomial, we can determine a polynomial of degree \( d \) where \( d \leq k - 1 \).
Two Common Types

### Polynomial Construction
- Share(s): \( n \) points on the polynomial
- Based in Lagrange’s interpolation theorem

### Theorem
Given \( k \) distinct points on a polynomial, we can determine a polynomial of degree \( d \) where \( d \leq k - 1 \).

- I.e., \( t = k \)
Two Common Types

Polynomial Construction

- Share(s): \( n \) points on the polynomial
- Based in Lagrange’s interpolation theorem

Theorem

Given \( k \) distinct points on a polynomial, we can determine a polynomial of degree \( d \) where \( d \leq k - 1 \).

- I.e., \( t = k \)
- Constant term necessary to reconstruct the secret
  - Shamir’s Secret Sharing Algorithm
Two Common Types

Polynomial Construction
- Share(s): $n$ points on the polynomial
- Based in Lagrange’s interpolation theorem

Theorem
Given $k$ distinct points on a polynomial, we can determine a polynomial of degree $d$ where $d \leq k - 1$.

- I.e., $t = k$
- Constant term necessary to reconstruct the secret
  - Shamir’s Secret Sharing Algorithm

Additive Construction
- Share(s): $n$ numbers adding up to an encoding of $s$
Two Common Types

Polynomial Construction
- Share(s): $n$ points on the polynomial
- Based in Lagrange’s interpolation theorem

Theorem
Given $k$ distinct points on a polynomial, we can determine a polynomial of degree $d$ where $d \leq k - 1$.
- I.e., $t = k$
- Constant term necessary to reconstruct the secret
  - Shamir’s Secret Sharing Algorithm

Additive Construction
- Share(s): $n$ numbers adding up to an encoding of $s$
- Requires all $n$ people to come together
Please flip over the cards we handed out, in order!
What’s the secret??

1. 04.41.56.54.01.16
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
3. 11.56.83.58.73.21
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
3. 11.56.83.58.73.21
4. 15.95.24.10.76.80
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
3. 11.56.83.58.73.21
4. 15.95.24.10.76.80
5. 08.90.61.60.43.66
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
3. 11.56.83.58.73.21
4. 15.95.24.10.76.80
5. 08.90.61.60.43.66
6. 27.80.77.16.20.77
What’s the secret??

1 04.41.56.54.01.16
2 12.17.70.77.54.23
3 11.56.83.58.73.21
4 15.95.24.10.76.80
5 08.90.61.60.43.66
6 27.80.77.16.20.77

Sum:
80.82.73.77.69.83
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
3. 11.56.83.58.73.21
4. 15.95.24.10.76.80
5. 08.90.61.60.43.66
6. 27.80.77.16.20.77

Sum:
80.82.73.77.69.83

<table>
<thead>
<tr>
<th>Letter</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>65</td>
</tr>
<tr>
<td>B</td>
<td>66</td>
</tr>
<tr>
<td>C</td>
<td>67</td>
</tr>
<tr>
<td>D</td>
<td>68</td>
</tr>
<tr>
<td>E</td>
<td>69</td>
</tr>
<tr>
<td>F</td>
<td>70</td>
</tr>
<tr>
<td>G</td>
<td>71</td>
</tr>
<tr>
<td>H</td>
<td>72</td>
</tr>
<tr>
<td>I</td>
<td>73</td>
</tr>
<tr>
<td>J</td>
<td>74</td>
</tr>
<tr>
<td>K</td>
<td>75</td>
</tr>
<tr>
<td>L</td>
<td>76</td>
</tr>
<tr>
<td>M</td>
<td>77</td>
</tr>
<tr>
<td>N</td>
<td>78</td>
</tr>
<tr>
<td>O</td>
<td>79</td>
</tr>
<tr>
<td>P</td>
<td>80</td>
</tr>
<tr>
<td>Q</td>
<td>81</td>
</tr>
<tr>
<td>R</td>
<td>82</td>
</tr>
<tr>
<td>S</td>
<td>83</td>
</tr>
<tr>
<td>T</td>
<td>84</td>
</tr>
<tr>
<td>U</td>
<td>85</td>
</tr>
<tr>
<td>V</td>
<td>86</td>
</tr>
<tr>
<td>W</td>
<td>87</td>
</tr>
<tr>
<td>X</td>
<td>88</td>
</tr>
<tr>
<td>Y</td>
<td>89</td>
</tr>
<tr>
<td>Z</td>
<td>90</td>
</tr>
</tbody>
</table>
What’s the secret??

1. 04.41.56.54.01.16
2. 12.17.70.77.54.23
3. 11.56.83.58.73.21
4. 15.95.24.10.76.80
5. 08.90.61.60.43.66
6. 27.80.77.16.20.77

Sum:
80.82.73.77.69.83

<table>
<thead>
<tr>
<th>Letter</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>65</td>
</tr>
<tr>
<td>B</td>
<td>66</td>
</tr>
<tr>
<td>C</td>
<td>67</td>
</tr>
<tr>
<td>D</td>
<td>68</td>
</tr>
<tr>
<td>E</td>
<td>69</td>
</tr>
<tr>
<td>F</td>
<td>70</td>
</tr>
<tr>
<td>G</td>
<td>71</td>
</tr>
<tr>
<td>H</td>
<td>72</td>
</tr>
<tr>
<td>I</td>
<td>73</td>
</tr>
<tr>
<td>J</td>
<td>74</td>
</tr>
<tr>
<td>K</td>
<td>75</td>
</tr>
<tr>
<td>L</td>
<td>76</td>
</tr>
<tr>
<td>M</td>
<td>77</td>
</tr>
<tr>
<td>N</td>
<td>78</td>
</tr>
<tr>
<td>O</td>
<td>79</td>
</tr>
<tr>
<td>P</td>
<td>80</td>
</tr>
<tr>
<td>Q</td>
<td>81</td>
</tr>
<tr>
<td>R</td>
<td>82</td>
</tr>
<tr>
<td>S</td>
<td>83</td>
</tr>
<tr>
<td>T</td>
<td>84</td>
</tr>
<tr>
<td>U</td>
<td>85</td>
</tr>
<tr>
<td>V</td>
<td>86</td>
</tr>
<tr>
<td>W</td>
<td>87</td>
</tr>
<tr>
<td>X</td>
<td>88</td>
</tr>
<tr>
<td>Y</td>
<td>89</td>
</tr>
<tr>
<td>Z</td>
<td>90</td>
</tr>
</tbody>
</table>
What’s the secret??

<table>
<thead>
<tr>
<th>Letter</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>65</td>
</tr>
<tr>
<td>B</td>
<td>66</td>
</tr>
<tr>
<td>C</td>
<td>67</td>
</tr>
<tr>
<td>D</td>
<td>68</td>
</tr>
<tr>
<td>E</td>
<td>69</td>
</tr>
<tr>
<td>F</td>
<td>70</td>
</tr>
<tr>
<td>G</td>
<td>71</td>
</tr>
<tr>
<td>H</td>
<td>72</td>
</tr>
<tr>
<td>I</td>
<td>73</td>
</tr>
<tr>
<td>J</td>
<td>74</td>
</tr>
<tr>
<td>K</td>
<td>75</td>
</tr>
<tr>
<td>L</td>
<td>76</td>
</tr>
<tr>
<td>M</td>
<td>77</td>
</tr>
<tr>
<td>N</td>
<td>78</td>
</tr>
<tr>
<td>O</td>
<td>79</td>
</tr>
<tr>
<td>P</td>
<td>80</td>
</tr>
<tr>
<td>Q</td>
<td>81</td>
</tr>
<tr>
<td>R</td>
<td>82</td>
</tr>
<tr>
<td>S</td>
<td>83</td>
</tr>
<tr>
<td>T</td>
<td>84</td>
</tr>
<tr>
<td>U</td>
<td>85</td>
</tr>
<tr>
<td>V</td>
<td>86</td>
</tr>
<tr>
<td>W</td>
<td>87</td>
</tr>
<tr>
<td>X</td>
<td>88</td>
</tr>
<tr>
<td>Y</td>
<td>89</td>
</tr>
<tr>
<td>Z</td>
<td>90</td>
</tr>
</tbody>
</table>

Sum:
80.82.73.77.69.83

PRIMES!
How to share your secrets?

- Secure Communication
  - Secret Key Encryption
  - Public Key Encryption
- Secret Sharing
  - Shamir’s Secret Sharing Algorithm
Acknowledgements

We would like to thank...

- ...our PRIMES mentors Lalita Devadas and Alexandra Henzinger,
- ...our parents,
- ...and the PRIMES coordinators for this amazing opportunity!