Cryptography: The Landscape, Fundamental Primitives, and Security

David Brumley
dbrumley@cmu.edu
Carnegie Mellon University
The Landscape

Jargon in Cryptography
Good News: OTP has perfect secrecy

**Thm:** The One Time Pad is Perfectly Secure

Must show: \( \Pr[ E(k, m_0) = c ] = \Pr[ E(k, m_1) = c ] \)

where \( |M| = \{0,1\}^m \)

**Proof:**

\[
\begin{align*}
\Pr[E(k, m_0) = c] &= \Pr[k \oplus m_0 = c] \\
&= \frac{|k \in \{0,1\}^m : k \oplus m_0 = c|}{\{0,1\}^m} \\
&= \frac{1}{2^m} \\
\Pr[E(k, m_1) = c] &= \Pr[k \oplus m_1 = c] \\
&= \frac{|k \in \{0,1\}^m : k \oplus m_1 = c|}{\{0,1\}^m} \\
&= \frac{1}{2^m}
\end{align*}
\]

Therefore, \( \Pr[E(k, m_0) = c] = \Pr[E(k, m_1) = c] \)
The “Bad News” Theorem

Theorem: Perfect secrecy requires $|K| \geq |M|$
Kerckhoffs’ Principle

The system must be *practically*, if not mathematically, indecipherable

\[
\Pr \left[ \text{ } \right] \ll (\text{Key length})
\]

- Security is only preserved against efficient adversaries running in (probabilistic) polynomial time (PPT) and space
- Adversaries can succeed with some small probability (that is small enough it is hopefully not a concern)
  - Ex: Probability of guessing a password

“A scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability”
The Landscape

\[ f(x) = y \]

\[ F: x \rightarrow y \]

\[ P_i: x \rightarrow x \]

Secrecy

"Application"
Pseudorandom Number Generators

Amplify small amount of randomness to large “pseudo-random” number with a 
*pseudo-random number generator* (PNRG)

\[ 0, 1^* \]

Let \( S : \{0, 1\}^s \) and \( K : \{0, 1\}^k \)

\[ G : S \rightarrow K \text{ where } k \gg s \]
**One Way Functions**

**Defn**: A function \( f \) is one-way if:

1. \( f \) can be computed in polynomial time
2. No polynomial time adversary \( A \) can invert with more than negligible probability

\[
\Pr[f(A(f(x))) = f(x)] < \epsilon
\]

**Note**: mathematically, a function is one-way if it is not one-to-one. Here we mean something stronger.
Candidate One-Way Functions

• Factorization. Let $N = p \times q$, where $|p| = |q| = |N|/2$. We believe factoring $N$ is hard.

• Discrete Log. Let $p$ be a prime, $x$ be a number between 0 and $p$. Given $g^x \mod p$, it is believed hard to recover $x$. 

$$0 < x < p - 1$$
The relationship

PRNG exist $\iff$ OWF exist
Thinking About Functions

A function is just a mapping from inputs to outputs:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f_1(x)$</th>
<th>$f_2(x)$</th>
<th>$f_3(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

Which function is not random?
Thinking About Functions

A function is just a mapping from inputs to outputs:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f_1(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f_2(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f_3(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

... 

What is random is the way we pick a function
Game-based Interpretation

Random Function

<table>
<thead>
<tr>
<th>x</th>
<th>f_1(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

Query x=3

Query f(x)=2

Fill in random value

Note asking x=1, 2, 3, ... gives us our OTP randomness.
Pseudo Random Function (PRF) defined over $(K,X,Y)$:

$$F : K \times X \rightarrow Y$$

such that there exists an “efficient” algorithm to evaluate $F(k,x)$.
Pseudorandom functions are not to be confused with pseudorandom generators (PRGs). The guarantee of a PRG is that a single output appears random if the input was chosen at random. On the other hand, the guarantee of a PRF is that all its outputs appear random, regardless of how the corresponding inputs were chosen, as long as the function was drawn at random from the PRF family.

- wikipedia
PRNG exist $\iff$ OWF exist $\iff$ PRF exists
Abstractly: PRPs

Pseudo Random Permutation (PRP) defined over \( (K,X) \)

\[
E : K \times X \to X
\]

such that:
1. Exists “efficient” deterministic algorithm to evaluate \( E(k,x) \)
2. The function \( E(k, \cdot) \) is one-to-one
3. Exists “efficient” inversion algorithm \( D(k,y) \)

\[ f(k) = x \iff f(x') = y \]
Running example

• Example PRPs: 3DES, AES, ...

  AES: $K \times X \to X$ where $K = X = \{0, 1\}^{128}$

  3DES: $K \times X \to X$ where $X = \{0, 1\}^{64}$, $K = \{0, 1\}^{168}$

• Functionally, any PRP is also a PRF.
  - PRP is a PRF when $X = Y$ and is efficiently invertible
Security and Indistinguishability
Kerckhoffs’ Principle

The system must be *practically*, if not mathematically, indecipherable

- Security is only preserved against efficient adversaries running in polynomial time and space
- Adversaries can succeed with some small probability (that is small enough it is hopefully not a concern)
  - Ex: Probability of guessing a password

“A scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability”
A Practical OTP

\[ c = E(k, m) = m \oplus G(k) \]

\[ D(k, c) = c \oplus G(k) \]
Question

Can a PRNG-based pad have perfect secrecy?
1. Yes, if the PRNG is secure
2. No, there are no ciphers with perfect secrecy
3. No, the key size is shorter than the message
PRG Security

Recall PRNG:
Let $S : \{0, 1\}^s$ and $K : \{0, 1\}^k$

$G : S \rightarrow K$ where $k \gg s$

One requirement: Output of PRG is unpredictable (mimics a perfect source of randomness)

It should be impossible for any Alg to predict bit $i+1$ given the first $i$ bits:

$\exists i. G(k)|_{1,\ldots,i}^{\text{Alg}} \rightarrow G(k)|_{i+1,\ldots,n}$

Even predicting 1 bit is insecure
Suppose PRG is predictable:

$$\exists i. G(k)|_{1,\ldots,i} \xrightarrow{\text{Alg}} G(k)|_{i+1,\ldots,n}$$

Example
Adversarial Indistinguishability Game

Challenger:
I have a secure PRF. It’s just like real randomness!

I am any adversary. You can’t fool me.
Secure PRF: The Intuition

PRF

Real Random Function

Barrier

A
PRF Security Game
(A behavioral model)

World 0

E

1. Picks x
2. if(tbl[x] undefined) tbl[x] = rand() return y = tbl[x]
3. Guess and output b'

A

y = PRF(x)

World 1

E

1. Picks x
3. Outputs guess for b

A

For b=0,1: $W_b := \{\text{event that } A(W_b) = 1\}$

$\text{Adv}_{SS}[A,E] := | \Pr[W_0] - \Pr[W_1] | \ll \in [0,1]$

A doesn’t know which world he is in, but wants to figure it out.

Always 1
Secure PRF: An Alternate Interpretation

For $b = 0, 1$ define experiment $EXP(b)$ as:

$b \in \{0, 1\}$

$b = 0 : k \leftarrow K, f \leftarrow F(k, \cdot)$

$b = 1 : f \leftarrow \text{Random Function}$

$x_1, x_2, \ldots, x_q \in X$

$f(x_1), f(x_2), \ldots, f(x_q)$

Def: PRF is a secure PRF if for all efficient $A$:

$$\text{Adv}_{PRF}[A, F] := |\Pr[Exp(0) = 1] - \Pr[Exp(1) = 1]| < \epsilon$$

$(t, \varepsilon, \delta)$-Secure
Quiz

Let $F : K \times X \rightarrow \{0, 1\}^{128}$ be a secure PRF.
Is the following $G$ a secure PRF?

$$G(k, x) = \begin{cases} 
0^{128} & \text{if } x = 0 \\
F(k, x) & \text{otherwise}
\end{cases}$$

- No, it is easy to distinguish $G$ from a random function
- Yes, an attack on $G$ would also break $F$
- It depends on $F$
Semantic Security of Ciphers
What is a secure cipher?

Attackers goal: recover one plaintext (for now)

Attempt #1: Attacker cannot recover key $E(\cdot, \cdot)$

*Insufficient:* $E(k,m) = m$

Attempt #2: Attacker cannot recover all of plaintext

$E(k,m) = E(k,m_0 \ldots m_{x-1})$ and $E(k,m_0 || m_1) = m_0 || E(k,m_1)$

*Insufficient:* $E(k,m_0 || m_1) = m_0 || E(k,m_1)$

Recall Shannon’s Intuition: $c$ should reveal no information about $m$
Adversarial Indistinguishability Game

Challenger: I have a secure cipher E

A

I am any adversary. I can break your crypto.
Semantic Security Motivation

1. \( A \) sends \( m_0, m_1 \) s.t. \(|m_0| = |m_1|\) to the challenger.

2. Challenger encrypts one at random. Sends back \( c \).

3. \( A \) tries to guess which message was encrypted.

4. Challenger wins if \( A \) is no better than guessing.

Semantically secure
Semantic Security Game

1. Picks $m_0, m_1$, $|m_0| = |m_1|$
2. Pick $b=0$
3. $k = \text{KeyGen}(l)$
4. $c = E(k, m_b)$
5. Guess and output $b'$

World 0

1. Picks $m_0, m_1$, $|m_0| = |m_1|$
2. Pick $b=1$
3. $k = \text{KeyGen}(l)$
4. $c = E(k, m_b)$
5. Guess and output $b'$

World 1

A doesn’t know which world he is in, but wants to figure it out.

Semantic security is a behavioral model getting at any $A$ behaving the same in either world when $E$ is secure.
Semantic Security Game

(A behavioral model)

World 0

1. Picks \( m_0, m_1, \quad |m_0| = |m_1| \)

2. Pick \( b=0 \)

3. \( k = \text{KeyGen}(l) \)

4. \( c = E(k,m_b) \)

5. Guess and output \( b' \)

World 1

1. Picks \( m_0, m_1, \quad |m_0| = |m_1| \)

2. Pick \( b=1 \)

3. \( k = \text{KeyGen}(l) \)

4. \( c = E(k,m_b) \)

5. Guess and output \( b' \)

A doesn’t know which world he is in, but wants to figure it out.

For \( b=0,1 \): \( W_b := [ \text{event that } A(W_b) = 1 ] \)

\[ \text{Adv}_{\text{SS}}[A,E] := | \text{Pr}[W_0] - \text{Pr}[W_1] | \in [0,1] \]
Example 1: Guessing

1. **World 0**
   - Pick $b = 0$
   - $k = \text{KeyGen}(l)$
   - $c = E(k, m_b)$
   - $m_0, m_1, |m_0| = |m_1|$
   - Guess and output $b'$

2. **World 1**
   - Pick $b = 1$
   - $k = \text{KeyGen}(l)$
   - $c = E(k, m_b)$
   - $m_0, m_1, |m_0| = |m_1|$
   - Guess and output $b'$

**A guesses.** $W_b := \{\text{event that } A(W_b) = 1\}$. So $W_0 = 0.5$, and $W_1 = 0.5$

$$\text{Adv}_{SS}[A,E] := |0.5 - 0.5| = 0$$
Example 1: \( A \) is right 75% of time

\[ A \text{ guesses. } W_b := [ \text{event that } A(W_b) = 1 ]. \text{ So } W_0 = .25, \text{ and } W_1 = .75 \]

\[ \text{Adv}_{\text{SS}}[A,E] := | .25 - .75 | = .5 \]
**Example 1: A is right 25% of time**

1. Picks $m_0, m_1$, $|m_0| = |m_1|$  
2. Pick $b=0$  
3. $k=\text{KeyGen}(l)$  
4. $c = E(k, m_b)$  
5. Guess and output $b'$

**World 0**

1. Picks $m_0, m_1$, $|m_0| = |m_1|$  
2. Pick $b=0$  
3. $k=\text{KeyGen}(l)$  
4. $c = E(k, m_b)$  
5. Guess and output $b'$

**World 1**

1. Picks $m_0, m_1$, $|m_0| = |m_1|$  
2. Pick $b=1$  
3. $k=\text{KeyGen}(l)$  
4. $c = E(k, m_b)$  
5. Guess and output $b'$

$A$ guesses. $W_b := \{ \text{event that } A(W_b) = 1 \}$. So $W_0 = .75$, and $W_1 = .25$

$\text{Adv}_{SS}[A,E] := | .75 - .25 | = .5$

Note for $W_0$, $A$ is wrong more often than right. $A$ should switch guesses.
**Semantic Security**

*Given:*
For $b=0,1$: $W_b := [\text{event that } A(W_b) = 1 ]$

$\text{Adv}_SS[A,E] := | \Pr[W_0] - \Pr[W_1] | \in [0,1]$

*Defn:*
$E$ is *semantically secure* if for all efficient $A$:

$\text{Adv}_SS[A,E]$ is negligible.

$\Rightarrow$ for all explicit $m_0, m_1 \in M$:

$\{ E(k,m_0) \} \approx_p \{ E(k,m_1) \}$

This is what it means to be secure against eavesdroppers. No partial information is leaked.
Something we believe is hard, e.g., factoring

Problem $A$

Easier

Problem $B$

Harder

Something we want to show is hard.
Proving Security
Security Reductions

**Reduction**: Problem $A$ is at least as hard as $B$ if an algorithm for solving $A$ efficiently (if it existed) could also be used as a subroutine to solve problem $B$ efficiently.

**Crux**: We don’t believe $A$ exists, so $B$ must be secure (contra-positive proof technique)
Example

*Reduction*: Problem **Factoring (A)** is at least as hard as **RSA (B)** if an algorithm for solving **Factoring (A)** efficiently (if it existed) could also be used as a subroutine to solve problem **RSA (B)** efficiently.
What’s *unknown*...

**Reduction**: Problem **RSA (A)** is at least as hard as **Factoring (B)** if an algorithm for solving **RSA (A)** efficiently (if it existed) could also be used as a subroutine to solve problem **Factoring (B)** efficiently.

- Synthesize p,q from just c, m, and N?
Guess It!
1. \(m = 1 \ldots 10\)
2. \(k = \text{KeyGen}(l)\)
3. \(c = E(k, m)\)

\[\text{bet} = ? = m\]

Suppose \(A\) is in a guessing game. Guess It! uses \(E\) to encrypt. How can we prove, in this setting, that \(E\) is secure?

**Reduction:** If \(A\) does better than 1/10, we break \(E\) in the semantic security game. Showing security of \(E\) reduces to showing if \(A\) exists, it could break the semantic security game. (Equivalently, if \(E\) is semantically secure, then the probability \(A\) wins is at most 10%.)

**Note:** The “type” of \(A\) is \(A: c \rightarrow \text{bet}\)
Idea

Reduction: We build an adversary $B$ that uses $A$ as a subroutine. Our adversary $B$ has the property if $A$ wins at Guess It! with probability significantly greater than 10%, $B$ will have a non-negligible advantage in our semantic security game.

– If $E$ secure, Guess It! is secure.
– Equivalently, if Guess It! insecure, $E$ is insecure
In the real version, $A$ always gets an encryption of the real message.

- $\Pr[A \text{ wins in real version}] = p_0$
In the ideal version, $A$ always gets an encryption of a constant, say 1. ($A$ still only wins if it gets $m$ correct.)

- $\Pr[A$ wins in Idealized Version$] = p_1 = 1/10$
Reduction

If B is in world 0, then \( \Pr[b' = 1] = p_0 \)
  - B can guess \( r == \text{bet} \) with prob. \( p_0 \).
If B is in world 1, then \( \Pr[b' = 1] = p_1 = 1/10 \)

For \( b=0,1 \): \( W_b := \) [ event that \( B(W_b) = 1 \) ]
\( \text{Adv}_{SS}[A,E] = | \Pr[ W_0 ] - \Pr[ W_1 ] | \)
\( = |p_0 - p_1| \)
Reduction

World $b = \{0, 1\}$

- If $B$ is in world 0, then $\Pr[b' = 1] = p_0$  
  - $B$ can guess $r == \text{bet}$ with prob. $p_0$.
- If $B$ is in world 1, then $\Pr[b' = 1] = p_1 = 1/10$
- For $b=0,1$: $W_b := $ [event that $B(W_b) = 1$ ]
- Advantage:
  $$\text{Adv}_{SS}[A, E] = \left| \Pr[W_0] - \Pr[W_1] \right| = |p_0 - p_1|$$

$E$

2. $m_b = b$
3. $k = \text{KeyGen}(l)$
4. $c = E(k, m_b)$

$m_0, m_1$

$B$

$r = \text{random } 1, \ldots, 10$
$m_0 = r$
$m_1 = 1 \text{ (const)}$

$A$

$\downarrow \text{bet}$

$b' = (r == \text{bet})$

Suppose 33% correct

33%-%10 = 23% Advantage
Reduction Example 2

Suppose efficient $A$ can always deduce LSB of PT from CT. Then $E = (E, D)$ is not semantically secure.

\[ \text{Adv}_{SS}[A, E] = | \Pr[W_0] - \Pr[W_1] | = |0 - 1| = 1 \]
Questions?
Thought
The “Bad News” Theorem

**Theorem**: Perfect secrecy requires $|K| \geq |M|$

In practice, we usually shoot for **computational security**.

And what about integrity and authenticity?
Secure PRF: Definition

- For $b = 0, 1$ define experiment $\text{EXP}(b)$ as:

  - Def: $F$ is a secure PRF if for all “efficient” A: $\text{Adv}_F[A, F] := \left| Pr[\text{EXP}(0) = 1] - Pr[\text{EXP}(1) = 1] \right|$ is “negligible”.

- $\text{Adv}_F[A, F]$ is defined as:
  - Chal. $b \in \{0, 1\}$
  - Adv. A $b' \in \{0, 1\}$
  - $b = 0 : k \leftarrow K, f \leftarrow F(k, \cdot)$
  - $b = 1 : f \leftarrow \text{Funs}[X, Y]$
  - $x_1, x_2, \ldots, x_q \in X$
  - $f(x_1), f(x_2), \ldots, f(x_q)$
Let $F : K \times X \rightarrow \{0, 1\}^{128}$ be a secure PRF. Is the following $G$ a secure PRF?

$$G(k, x) = \begin{cases} 
0^{128} & \text{if } x = 0 \\
F(k, x) & \text{otherwise}
\end{cases}$$

- No, it is easy to distinguish $G$ from a random function
- Yes, an attack on $G$ would also break $F$
- It depends on $F$
Secure PRPs (secure block cipher)

- Let \( E : K \times X \rightarrow Y \) be a PRP\((X = Y)\)

\[
\begin{align*}
Perms[X] & : \text{the set of all one-to-one functions from } X \text{ to } Y \\
S_F &= \{ E(k, \cdot) \text{ s.t. } k \in K \} \subseteq Perms[X]
\end{align*}
\]

- Intuition: a PRP is \textbf{secure} if
  A random function in \( Perms[X] \) is indistinguishable from a random function in \( S_F \)
Secure PRP: (secure block cipher)

- For $b = 0, 1$ define experiment $\exp(b)$ as:

  $b \in \{0, 1\}$

  $b = 0 : k \leftarrow K, f \leftarrow E(k, \cdot)$

  $b = 1 : f \leftarrow \text{Perms}[X]$

  $x_1, x_2, \ldots, x_q \in X$

  $f(x_1), f(x_2), \ldots, f(x_q)$

- **Def:** $E$ is a secure PRP if for all “efficient” $A$:

  $\text{Adv}_{\text{PRP}}[A, E] := |Pr[\exp(0) = 1] - Pr[\exp(1) = 1]|$

  is “negligible”.

$\exp(b)$
Modern Notions: Indistinguishability and Semantic Security