Stream ciphers

The One Time Pad
Symmetric Ciphers: definition

Def: a cipher defined over \((K, M, C)\) is a pair of “efficient” algs \((E, D)\) where

\[
E : K \times M \to C, \quad D : K \times C \to M
\]

s.t. \( \forall m \in M, \ k \in K : D(k, E(k, m)) = m \)

• \( E \) is often randomized. \( D \) is always deterministic.
The One Time Pad  
(Vernam 1917)

First example of a “secure” cipher

\[ M = C = \{0,1 \}^n, \quad K = \{0,1 \}^n \]

key = (random bit string as long the message)
The One Time Pad

(Vernam 1917)

\[ C := E(k, m) = k \oplus m \]

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

Indeed:

\[ D(k, E(k, m)) = D(k, k \oplus m) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m = m \]

msg: 0 1 1 0 1 1 1
key: 1 0 1 1 0 1 0
CT: \[ \oplus \]
You are given a message \( (m) \) and its OTP encryption \( (c) \).

Can you compute the OTP key from \( m \) and \( c \) ?

No, I cannot compute the key.

Yes, the key is \( k = m \oplus c \).

I can only compute half the bits of the key.

Yes, the key is \( k = m \oplus m \).
The One Time Pad (Vernam 1917)

Very fast enc/dec !!
   ... but long keys (as long as plaintext)

Is the OTP secure? What is a secure cipher?
What is a secure cipher?

Attacker’s abilities: \textbf{CT only attack} (for now)

Possible security requirements:

attempt #1: \textcolor{red}{attacker cannot recover secret key} \hfill \textcolor{red}{\color{red}{E(k, m) = m \text{ would be secure}}}

attempt #2: \textcolor{red}{attacker cannot recover all of plaintext} \hfill \textcolor{red}{\color{red}{E(k, m_0 || h) = m_0 || c \oplus m, \text{ would be secure}}}

Shannon’s idea:

\textbf{CT should reveal no “info” about PT}
Information Theoretic Security
(Shannon 1949)

Def: A cipher \((E, D)\) over \((\mathcal{K}, \mathcal{M}, \mathcal{C})\) has **perfect secrecy** if

\[
\forall m_0, m_1 \in \mathcal{M} \quad (\text{len}(m_0) = \text{len}(m_1)) \quad \text{and} \quad \forall c \in \mathcal{C}
\]

\[
\Pr[E(\kappa, m_0) = c] = \Pr[E(\kappa, m_1) = c]
\]

where \(\kappa\) is uniform on \(\mathcal{K}\) \((\kappa \leftarrow \mathcal{K})\)
**Def:** A cipher \((E,D)\) over \((K,M,C)\) has **perfect secrecy** if

\[
\forall m_0, m_1 \in M \ ( \ |m_0| = |m_1| ) \quad \text{and} \quad \forall c \in C
\]

\[
Pr[ E(k,m_0) = c ] = Pr[ E(k,m_1) = c ] \quad \text{where} \quad k \xleftarrow{} K
\]

⇒ Given \(CT\) can’t tell if msg is \(m_0\) or \(m_1\) (for all \(m_0, m_1\))

⇒ Most powerful adv. learns nothing about \(PT\) from \(CT\)

⇒ No \(CT\) only attack!! (but other attacks possible)
Lemma: OTP has perfect secrecy.

Proof:

\[ \forall m, c: \quad \Pr_{k \in \mathcal{K}} \left[ E(k, m) = c \right] = \frac{\# \{ k \in \mathcal{K} : E(k, m) = c \}}{|\mathcal{K}|} \]

So: \[ \forall m, c: \quad \# \{ k \in \mathcal{K} : E(k, m) = c \} = \text{const.} \]

\[ \implies \text{cipher has perfect secrecy} \]
Let \( m \in \mathcal{M} \) and \( c \in \mathcal{C} \).

How many OTP keys map \( m \) to \( c \)?

None

1

2

Depends on \( m \)
**Lemma:** OTP has perfect secrecy.

**Proof:**

For OTP: \[ \forall m, c : \text{if } E(k, m) = c \]

\[ \implies k \oplus m = c \implies k = m \oplus c \]

\[ \implies \#\{ k \in \mathcal{K} : E(k, m) = c \} = 1 \]

\[ \implies \text{OTP has perfect secrecy} \]
The bad news ...

Thm: perfect secrecy \(\Rightarrow\) \(|K| \geq |M|\)

i.e. perfect secrecy \(\Rightarrow\) key-len \(\geq\) msg-len

\(\Rightarrow\) hard to use in practice !!
End of Segment
Stream ciphers

Pseudorandom Generators
Review

Cipher over (K,M,C): a pair of “efficient” algs \((E, D)\) s.t.

\[
\forall m \in M, \; k \in K: \; D(k, E(k, m)) = m
\]

Weak ciphers: subs. cipher, Vigener, ...

A good cipher: \(\text{OTP} \quad M=C=K=\{0,1\}^n\)

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

Lemma: OTP has perfect secrecy (i.e. no CT only attacks)

Bad news: perfect-secrecy \(\Rightarrow\) key-len \(\geq\) msg-len
Stream Ciphers: making OTP practical

idea: replace “random” key by “pseudorandom” key

PRG is a function $G: \{0,1\}^s \rightarrow \{0,1\}^n$ $n \gg s$

(efi. computable by a deterministic algorithm)
Stream Ciphers: making OTP practical

\[ C := E(K, m) = m \oplus g(K) \]

\[ D(K, C) = C \oplus g(K) \]
Can a stream cipher have perfect secrecy?

- Yes, if the PRG is really “secure”
- No, there are no ciphers with perfect secrecy
- Yes, every cipher has perfect secrecy
- No, since the key is shorter than the message
Stream Ciphers: making OTP practical

Stream ciphers cannot have perfect secrecy!!

• Need a different definition of security

• Security will depend on specific PRG
PRG must be unpredictable

Suppose PRG is predictable:

\[
\exists i : \mathcal{G}(k)_{i, \ldots, i} \xrightarrow{\text{alg}} \mathcal{G}(k)_{i+1, \ldots, n}
\]

Then:

\[
\mathcal{C} \oplus \mathcal{M}
\]

\[
\mathcal{G}(k)_{\ldots, i} \xrightarrow{\text{alg}} \mathcal{G}(k)_{i+1, \ldots}
\]

is a problem!
We say that $G: K \rightarrow \{0,1\}^n$ is **predictable** if:

$$\exists \text{"eff" alg. } A \text{ and } \exists 0 \leq i \leq n-1 \text{ s.t.}$$

$$\Pr_{K \leftarrow \{0,1\}^k, i_1, \ldots, i_{i-1}} \left[ A(G(i_1, \ldots, i_{i-1})) = G(K) \mid i_{i+1} \right] > \frac{1}{2} + \varepsilon$$

For non-negligible $\varepsilon$ (e.g. $\varepsilon = \frac{1}{2^{20}}$)

**Def:** PRG is **unpredictable** if it is not predictable

$$\Rightarrow \forall i: \text{no "eff" adv. can predict bit } (i+1) \text{ for "non-neg" } \varepsilon$$
Suppose \( G:K \rightarrow \{0,1\}^n \) is such that for all \( k \): \( \text{XOR}(G(k)) = 1 \)

Is \( G \) predictable??

Yes, given the first bit I can predict the second
No, \( G \) is unpredictable
Yes, given the first \((n-1)\) bits I can predict the \( n \)'th bit
It depends
Weak PRGs (do not use for crypto)

Linear congruential generator with parameters \(a, b, p\): 

\[
\begin{align*}
    r[i] &\leftarrow a \cdot r[i-1] + b \mod p \\
    \text{output bits of } r[i] \\
    i &\leftarrow i + 1
\end{align*}
\]

\(seed = r[0]\)

glibc random(): 

\[
    r[i] \leftarrow (r[i-3] + r[i-31]) \mod 2^{32}
\]

output \(r[i] >> 1\)

*never use \texttt{random()} for crypto!! (e.g. Kerberos V4)*
End of Segment
Stream ciphers

Negligible vs. non-negligible
Negligible and non-negligible

- **In practice:** $\varepsilon$ is a scalar and
  - $\varepsilon$ non-neg: $\varepsilon \geq 1/2^{30}$ (likely to happen over 1GB of data)
  - $\varepsilon$ negligible: $\varepsilon \leq 1/2^{80}$ (won’t happen over life of key)

- **In theory:** $\varepsilon$ is a function $\varepsilon : \mathbb{Z}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ and
  - $\varepsilon$ non-neg: $\exists d: \varepsilon(\lambda) \geq 1/\lambda^d$ inf. often $\quad (\varepsilon \geq 1/poly, \text{ for many } \lambda)$
  - $\varepsilon$ negligible: $\forall d, \lambda \geq \lambda_d: \varepsilon(\lambda) \leq 1/\lambda^d \quad (\varepsilon \leq 1/poly, \text{ for large } \lambda)$
Few Examples

\[ \varepsilon(\lambda) = \frac{1}{2^\lambda} : \text{negligible} \]

\[ \varepsilon(\lambda) = \frac{1}{\lambda^{1000}} : \text{non-negligible} \]

\[ \varepsilon(\lambda) = \begin{cases} 
1/2^\lambda & \text{for odd } \lambda \\
1/\lambda^{1000} & \text{for even } \lambda 
\end{cases} \]

Negligible

Non-negligible
PRGs: the rigorous theory view

PRGs are “parameterized” by a security parameter $\lambda$

• PRG becomes “more secure” as $\lambda$ increases

Seed lengths and output lengths grow with $\lambda$

For every $\lambda=1,2,3,...$ there is a different PRG $G_{\lambda}$:

$$G_{\lambda}: K_{\lambda} \rightarrow \{0,1\}^{n(\lambda)}$$

(in the lectures we will always ignore $\lambda$)
An example asymptotic definition

We say that $G_\lambda : K_\lambda \rightarrow \{0,1\}^{n(\lambda)}$ is **predictable** at position $i$ if:

there exists a *polynomial* time (in $\lambda$) algorithm $A$ s.t.

$$\Pr_{k \leftarrow K_\lambda} \left[ A(\lambda, G_\lambda(k)_{1,\ldots,i}) = G_\lambda(k)_{i+1} \right] > 1/2 + \varepsilon(\lambda)$$

for some *non-negligible* function $\varepsilon(\lambda)$
End of Segment
Stream ciphers

Attacks on OTP and stream ciphers
OTP: \[ E(k,m) = m \oplus k \quad , \quad D(k,c) = c \oplus k \]

Making OTP practical using a PRG: \[ G: K \rightarrow \{0,1\}^n \]

Stream cipher: \[ E(k,m) = m \oplus G(k) \quad , \quad D(k,c) = c \oplus G(k) \]

Security: PRG must be unpredictable (better def in two segments)
Attack 1: **two time** pad is insecure !!

Never use stream cipher key more than once !!

\[
\begin{align*}
C_1 & \leftarrow m_1 \oplus \text{PRG}(k) \\
C_2 & \leftarrow m_2 \oplus \text{PRG}(k)
\end{align*}
\]

Eavesdropper does:

\[
C_1 \oplus C_2 \rightarrow
\]

Enough redundancy in English and ASCII encoding that:

\[
m_1 \oplus m_2 \rightarrow m_1, m_2
\]
Real world examples

• Project Venona

• MS-PPTP (windows NT):

\[ \left[ m_1 \| m_2 \| m_3 \right] \oplus g(k) \]

\[ k = (k_{C \rightarrow S}, k_{S \rightarrow C}) \]

\[ \left[ s_1 \| s_2 \| s_3 \right] \oplus g(k) \]

Need different keys for \( C \rightarrow S \) and \( S \rightarrow C \)
802.11b WEP:

Length of IV: 24 bits

- Repeated IV after $2^{24} \approx 16M$ frames
- On some 802.11 cards: IV resets to 0 after power cycle
Avoid related keys

802.11b WEP:

key for frame #1:  \((1 \mathbin{||} k)\)
key for frame #2:  \((2 \mathbin{||} k)\)
A better construction

⇒ now each frame has a pseudorandom key

better solution: use stronger encryption method (as in WPA2)
Yet another example: disk encryption

1. To: Bob
   
2. Later:

3. To: Eve

Enc. disk

Only change
Two time pad: summary

Never use stream cipher key more than once !!

• Network traffic: negotiate new key for every session (e.g. TLS)

• Disk encryption: typically do not use a stream cipher
Attack 2: no integrity \((\text{OTP is malleable})\)

Modifications to ciphertext are undetected and have \textbf{predictable} impact on plaintext.
Attack 2: no integrity (OTP is malleable)

Modifications to ciphertext are undetected and have predictable impact on plaintext
End of Segment
Stream ciphers

Real-world Stream Ciphers
Old example (software): RC4 (1987)

- Used in HTTPS and WEP

- Weaknesses:
  1. Bias in initial output: \( \Pr[\text{2nd byte }= 0] = \frac{2}{256} \)
  2. Prob. of \((0,0)\) is \( \frac{1}{256^2} + \frac{1}{256^3} \)
  3. Related key attacks
Old example (hardware): CSS (badly broken)

Linear feedback shift register (LFSR):

DVD encryption (CSS): 2 LFSRs
GSM encryption (A5/1,2): 3 LFSRs
Bluetooth (E0): 4 LFSRs

all broken
Old example (hardware): CSS (badly broken)

CSS: seed = 5 bytes = 40 bits

Easy to break in time $\approx 2^{17}$
Cryptanalysis of CSS

(2^{17} time attack)

For all possible initial settings of 17-bit LFSR do:
- Run 17-bit LFSR to get 20 bytes of output
- Subtract from CSS prefix $\Rightarrow$ candidate 20 bytes output of 25-bit LFSR
- If consistent with 25-bit LFSR, found correct initial settings of both !!

Using key, generate entire CSS output
Modern stream ciphers: eStream

PRG: \[ \{0,1\}^s \times R \rightarrow \{0,1\}^n \]

Nonce: a non-repeating value for a given key.

\[ E(k, m; r) = m \oplus \text{PRG}(k; r) \]

The pair \((k,r)\) is never used more than once.
eStream: Salsa 20 (SW+HW)

Salsa20: $\{0,1\}^{128}$ or $2^{256} \times \{0,1\}^{64} \rightarrow \{0,1\}^n$ (max $n = 2^{73}$ bits)

Salsa20($k; r$) := $H(k, (r, 0)) \parallel H(k, (r, 1)) \parallel ...$

h: invertible function. designed to be fast on x86 (SSE2)
Is Salsa20 secure (unpredictable)?

- Unknown: no known **provably** secure PRGs
- In reality: no known attacks better than exhaustive search
Performance: Crypto++ 5.6.0 [ Wei Dai ]

AMD Opteron, 2.2 GHz (Linux)

<table>
<thead>
<tr>
<th>PRG</th>
<th>Speed (MB/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC4</td>
<td>126</td>
</tr>
</tbody>
</table>
| eStream
|                |
| Salsa20/12     | 643           |
| Sosemanuk      | 727           |
Generating Randomness  (e.g. keys, IV)

Pseudo random generators in practice:  (e.g. /dev/random)

- Continuously add entropy to internal state
- Entropy sources:
  - Timing:  hardware interrupts (keyboard, mouse)

NIST SP 800-90:  NIST approved generators
End of Segment
Stream ciphers

PRG Security Defs
Let $G:K \rightarrow \{0,1\}^n$ be a PRG.

Goal: define what it means that

\[ \left[ k \leftarrow G, \text{ output } G(k) \right] \]

is “indistinguishable” from

\[ \left[ r \leftarrow \{0,1\}^n, \text{ output } r \right] \]
Statistical Tests

**Statistical test** on \(\{0,1\}^n\):

an alg. \(A\) s.t. \(A(x)\) outputs “0” or “1”

Examples:

1. \(A(x)=1\) \iff \(\left| \#0(x) - \#1(x) \right| \leq 10 \cdot \sqrt{n} \)

2. \(A(x)=1\) \iff \(\left| \#00(x) - \frac{n}{4} \right| \leq 10 \cdot \sqrt{n} \)
Statistical Tests

More examples:

\[ A(x) = 1 \quad \text{if} \quad \max_{\text{run-of-0}}(x) < 10 \cdot \log_2(n) \]
Advantage

Let $G:K \rightarrow \{0,1\}^n$ be a PRG and $A$ a stat. test on $\{0,1\}^n$

Define:

$$\text{Adv}_{\text{PRG}}[A,G] = \left| \Pr_{k \leftarrow \mathcal{K}} [A(G(k)) = 1] - \Pr_{r \leftarrow \{0,1\}^n} [A(r) = 1] \right| \in [0,1]$$

Adv close to 1 $\Rightarrow$ $A$ can dist. $G$ from random
Adv close to 0 $\Rightarrow$ $A$ cannot

A silly example: $A(x) = 0$ $\Rightarrow$ $\text{Adv}_{\text{PRG}}[A,G] =$
Suppose \( G:K \rightarrow \{0,1\}^n \) satisfies \( \text{msb}(G(k)) = 1 \) for \( 2/3 \) of keys in \( K \)

Define stat. test \( A(x) \) as:

\[
\text{if } \text{msb}(x) = 1 \text{ then output "1" else output "0"}
\]

Then

\[
\text{Adv}_{\text{PRG}} [A,G] = \left| \text{Pr}[ A(G(k)) = 1] - \text{Pr}[ A(r) = 1] \right| = \frac{2/3}{\sqrt{2}} = \frac{1}{6}
\]
Def: We say that $G:K \rightarrow \{0,1\}^n$ is a **secure PRG** if

$$\forall \text{ "efficient" statistical tests } A:$$

$$\text{Adv}_{\text{prg}}[A,G] \text{ is "negligible"}$$

Are there provably secure PRGs?

but we have heuristic candidates.
Easy fact: a secure PRG is unpredictable

We show: PRG predictable $\Rightarrow$ PRG is insecure

Suppose $A$ is an efficient algorithm s.t.

$$\Pr_{\kappa \leftarrow \mathcal{R}} \left[ A(G(\kappa)\|\ldots\|\kappa) = G(\kappa)\|\ldots\|\kappa \right] > \frac{1}{2} + \varepsilon$$

for non-negligible $\varepsilon$ (e.g. $\varepsilon = 1/1000$)
Easy fact: a secure PRG is unpredictable

Define statistical test $B$ as:

$$B(x) = \begin{cases} 1 & \text{if } A(x_{1}, \ldots, i) = x_{i+1} \\ 0 & \text{else} \end{cases}$$

\[ r \leftarrow \{0, 1\}^n : \quad \Pr[B(r) = 1] = \frac{1}{2} \]

\[ r \leftarrow \mathcal{G} \quad \Pr[B(G(r)) = 1] > \frac{1}{2} + \varepsilon \]

\[ \Rightarrow \quad \text{Adv}_{PRG} [B, G] = \left| \Pr[B(r) = 1] - \Pr[B(G(r)) = 1] \right| > \varepsilon \]
Thm (Yao’82): an unpredictable PRG is secure

Let \( G:K \rightarrow \{0,1\}^n \) be PRG

“Thm”: if \( \forall i \in \{0, ... , n-1\} \) PRG \( G \) is unpredictable at pos. \( i \) then \( G \) is a secure PRG.

If next-bit predictors cannot distinguish \( G \) from random then no statistical test can !!
Let \( G:K \rightarrow \{0,1\}^n \) be a PRG such that from the last \( n/2 \) bits of \( G(k) \) it is easy to compute the first \( n/2 \) bits.

Is \( G \) predictable for some \( i \in \{0, \ldots, n-1\} \)?
More Generally

Let $P_1$ and $P_2$ be two distributions over $\{0,1\}^n$

Def: We say that $P_1$ and $P_2$ are

**computationally indistinguishable** (denoted $P_1 \approx_p P_2$)

if for all "efficient" statistical tests $A$

$$\left| \Pr_{x \leftarrow P_1} [A(x) = 1] - \Pr_{x \leftarrow P_2} [A(x) = 1] \right| < \text{negligible}$$

Example: a PRG is secure if

$$\{ k \overset{R}{\leftarrow} K : G(k) \} \approx_p \text{uniform}(\{0,1\}^n)$$
End of Segment
Stream ciphers

Semantic security

Goal: secure PRG $\Rightarrow$ “secure” stream cipher
What is a secure cipher?

Attacker’s abilities: obtains one ciphertext (for now)

Possible security requirements:

attempt #1: attacker cannot recover secret key

\[ E(k, m) = m \]

attempt #2: attacker cannot recover all of plaintext

\[ E(k, m_0 \parallel m_1) = m_0 \parallel m_1 \oplus K \]

Recall Shannon’s idea:

CT should reveal no “info” about PT
Recall Shannon’s perfect secrecy

Let \((E,D)\) be a cipher over \((K,M,C)\)

\((E,D)\) has perfect secrecy if \(\forall m_0, m_1 \in M \ (|m_0| = |m_1|)\)

\[\{ E(k,m_0) \} = \{ E(k,m_1) \} \quad \text{where} \quad k \leftarrow K\]

\((E,D)\) has perfect secrecy if \(\forall m_0, m_1 \in M \ (|m_0| = |m_1|)\)

\[\{ E(k,m_0) \} \approx_p \{ E(k,m_1) \} \quad \text{where} \quad k \leftarrow K\]

... but also need adversary to exhibit \(m_0, m_1 \in M\) explicitly
Semantic Security (one-time key)

For \( b=0,1 \) define experiments \( \text{EXP}(0) \) and \( \text{EXP}(1) \) as:

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

\[ \text{Adv}_{\text{SS}}[A,E] := \left| \Pr[W_0] - \Pr[W_1] \right| \in [0,1] \]
Semantic Security (one-time key)

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

$$Adv_{ss}[A,E] \text{ is negligible.}$$

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

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

Suppose efficient A can always deduce LSB of PT from CT.

$\Rightarrow \; E = (E,D)$ is not semantically secure.

Then $\text{Adv}_{SS}[B, E] = | \Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1] | = 1$. 
OTP is semantically secure

For all $A$: $\text{Adv}_{SS}[A, \text{OTP}] = \left| \Pr[A(k \oplus m_0) = 1] - \Pr[A(k \oplus m_1) = 1] \right|$
End of Segment
Stream ciphers

Stream ciphers are semantically secure

Goal: secure PRG $\Rightarrow$ semantically secure stream cipher
Stream ciphers are semantically secure

Thm: $G:K \rightarrow \{0,1\}^n$ is a secure PRG $\Rightarrow$

stream cipher $E$ derived from $G$ is sem. sec.

$\forall$ sem. sec. adversary $A$, $\exists$ a PRG adversary $B$ s.t.

$Adv_{SS}[A,E] \leq 2 \cdot Adv_{PRG}[B,G]$
Proof: intuition

\[
k \leftarrow K \quad \text{chal.}
\]
\[
ym_0, m_1 \quad \text{adv. A}
\]
\[
c \leftarrow m_0 \oplus G(k)
\]
\[
\downarrow \quad b' \approx 1
\]

\[
r \leftarrow \{0,1\}^n
\]
\[
ym_0, m_1 \quad \text{adv. A}
\]
\[
c \leftarrow m_0 \oplus r
\]
\[
\downarrow \quad b' \approx 1
\]

\[
k \leftarrow K \quad \text{chal.}
\]
\[
ym_0, m_1 \quad \text{adv. A}
\]
\[
c \leftarrow m_1 \oplus G(k)
\]
\[
\downarrow \quad b' \approx 1
\]

\[
r \leftarrow \{0,1\}^n
\]
\[
ym_0, m_1 \quad \text{adv. A}
\]
\[
c \leftarrow m_1 \oplus r
\]
\[
\downarrow \quad b' \approx 1
\]
Proof: Let $A$ be a semi-secure adversary.

For $b = 0, 1$:

$$W_b := \text{event that } b' = 1 +$$

$$\text{Adv}_{SS}[A, E] = \left| \Pr[W_0] - \Pr[W_1] \right|$$
Proof: Let $A$ be a semi-secure adversary.

\[
W_b := \text{event that } b' = 1 + \text{Adv}_{SS}[A, E] = \left| \Pr[W_0] - \Pr[W_1] \right|
\]

For $b=0,1$: $W_b := \text{event that } b' = 1$.

For $b=0,1$: $R_b := \text{event that } b' = 1$.
Proof: Let A be a semi. sec. adversary.

Claim 1: \( |\Pr[R_0] - \Pr[R_1]| = \text{Adv}_{SS}[A, \text{OTP}] = 0 \)

Claim 2: \( \exists B: |\Pr[W_b] - \Pr[R_b]| = \text{Adv}_{PRG}[B, \sigma] \quad \text{for } b=0,1 \)

\[ \Rightarrow \text{Adv}_{SS}[A,E] = |\Pr[W_0] - \Pr[W_1]| \leq 2 \cdot \text{Adv}_{PRG}[B,\sigma] \]
Proof of claim 2: \[ \exists B: \left| \Pr[W_0] - \Pr[R_0] \right| = \text{Adv}_{\text{PRG}}[B,G] \]

Algorithm B:

\[
\text{Adv}_{\text{PRG}}[B,G] = \left| \Pr_{r \leftarrow \{0,1\}^n}[B(r) = 1] - \Pr_{k \leftarrow \mathbb{G}}[B(\pi(k)) = 1] \right| = \left| \Pr[R_0] - \Pr[W_0] \right|
\]
End of Segment