Pseudorandom Generators Continued, Security Reductions

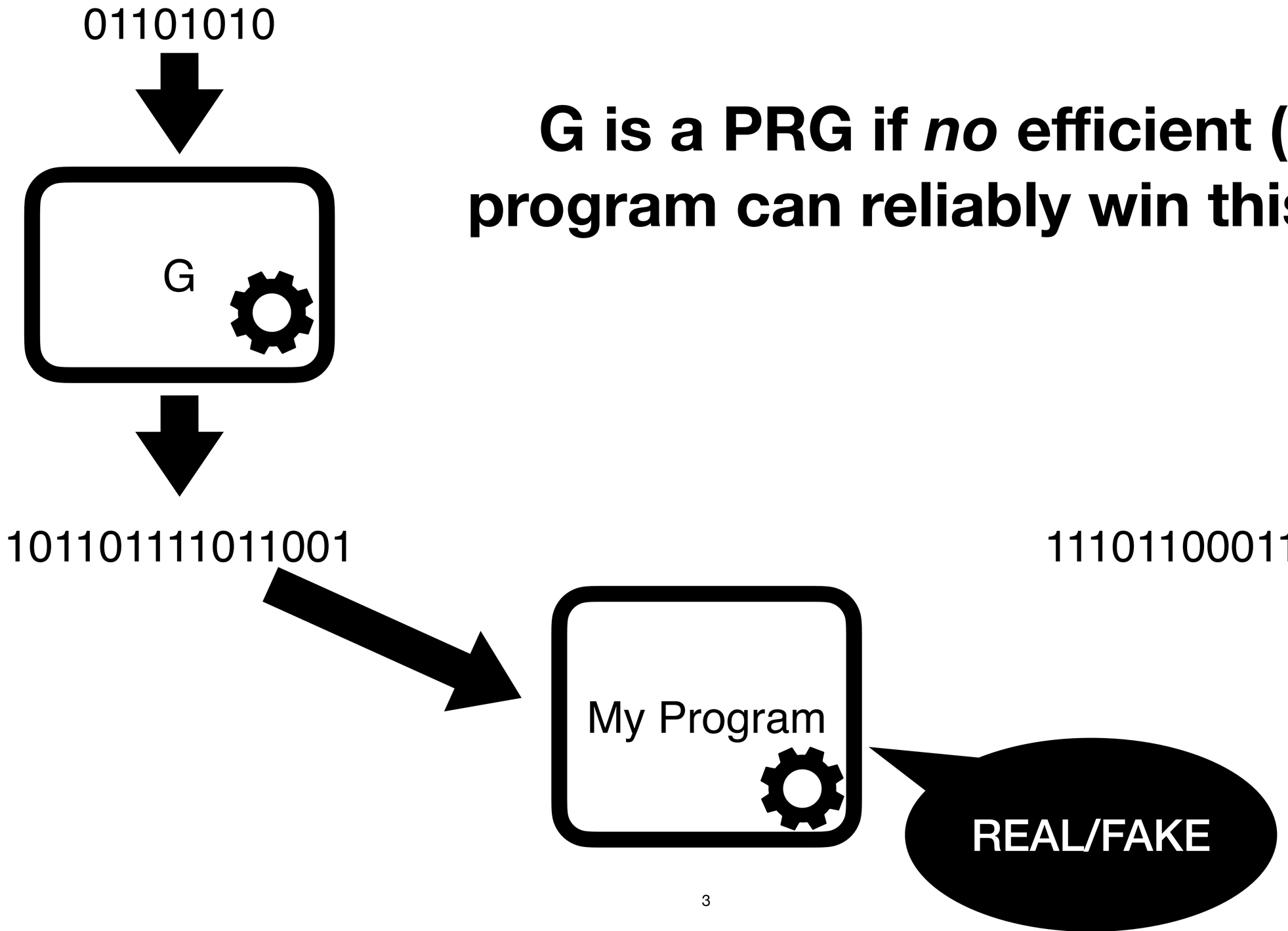
CS/ECE 407

Today’s objectives

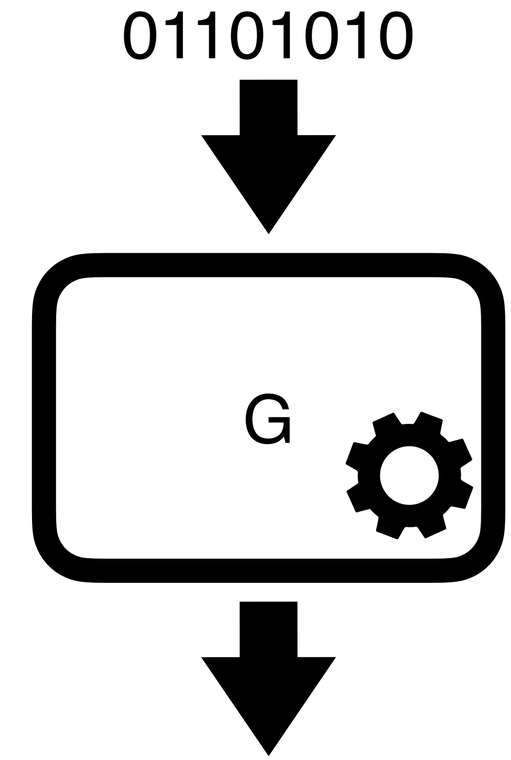
Construct PRG with long stretch from one with short stretch

Introduce the notion of a **security reduction**

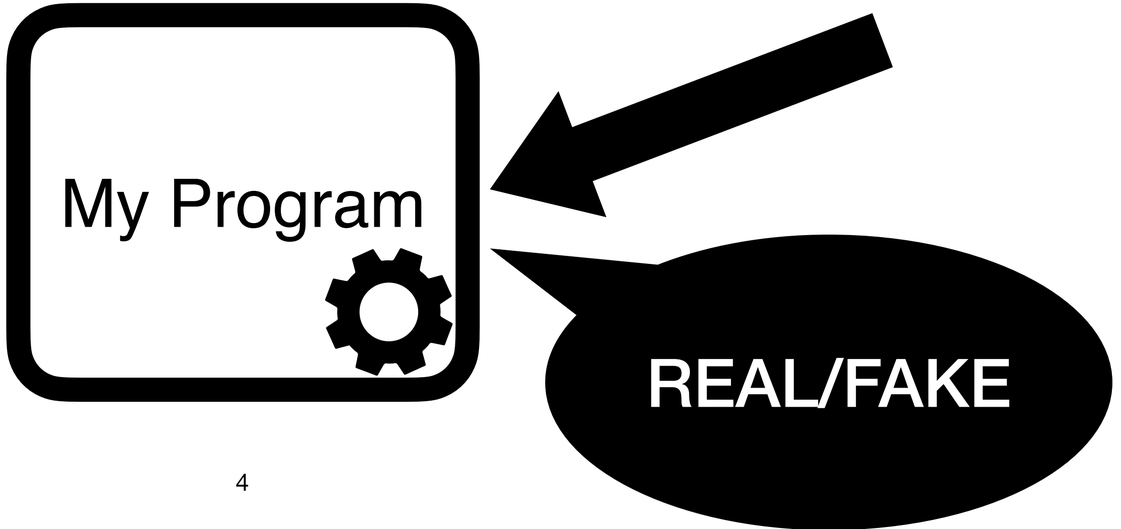
Understand an attack



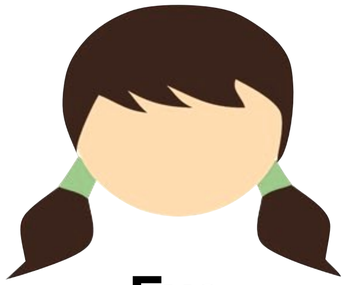
G is a PRG if *no* efficient (PPT) program can reliably win this game



G is a PRG if *no* efficient (PPT) program can reliably win this game



Alice



Eve

Bob

`k = 100101110`

`k = 100101110`

`G(k) = 001110110100110011001`

`G(k) = 001110110100110011001`

Indistinguishability

Let X, Y be two probability ensembles, and let A be an arbitrary (probabilistic) program that outputs 0 or 1. A's **advantage** is as follows:

$$\text{Advantage}_A(\lambda) = \left| \Pr\left[b = 1 \;\middle|\; \begin{array}{l} x \leftarrow_{\$} X_\lambda \\ b \leftarrow A(1^\lambda, x) \end{array} \right] - \Pr\left[b = 1 \;\middle|\; \begin{array}{l} y \leftarrow_{\$} Y_\lambda \\ b \leftarrow A(1^\lambda, y) \end{array} \right] \right|$$

We say that X, Y are **indistinguishable,** written $X \approx Y$ if for every polynomial-time program A:

$\text{Advantage}_A(\lambda)$ is negligible

best strategy is only negligibly better than guessing

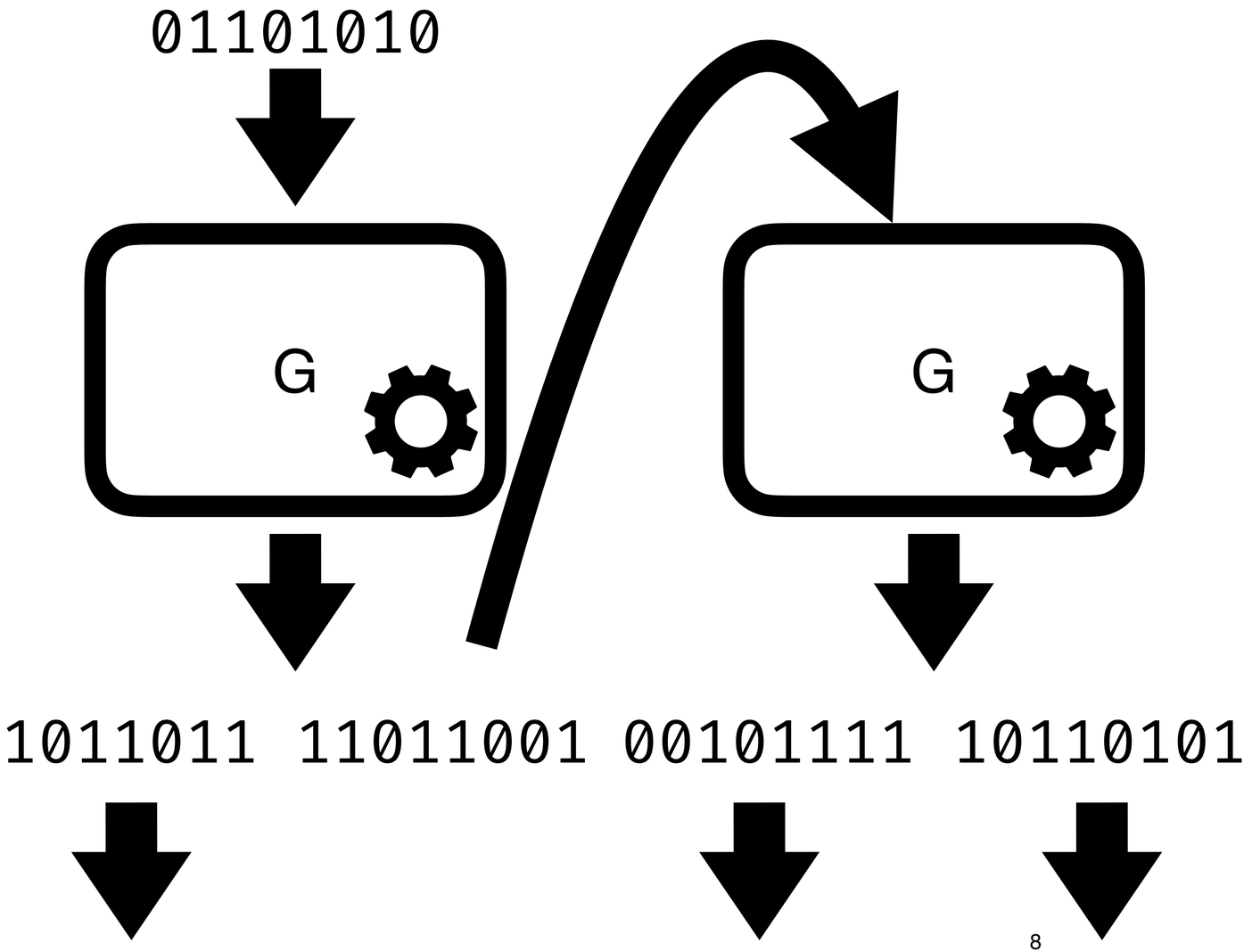
PRG security

Let G be a poly-time deterministic algorithm that on an input of length λ outputs a string of length $\lambda + s(\lambda)$.
G is a PRG if $s(\lambda)$ is always positive, and:

$$\left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\} \approx \left\{ r \;\middle|\; r \leftarrow \{0,1\}^{\lambda+s(\lambda)} \right\}$$

"If seed k is uniform and hidden, then $G(k)$ looks uniform"

Stretching the output of a PRG



This is a secure PRG

Security Reduction

$$G \text{ is a PRG} \implies G' \text{ is a PRG}$$

$$G \text{ is a not PRG} \impliedby G' \text{ is not a PRG}$$

Let G be a length-doubling PRG

$$G'(k) = \left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' = G(k) \\ r_1, r_2 = G(k') \end{array} \right\}$$

Claim: G' is a PRG

Let G be a length-doubling PRG

$$G'(k) = \left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' = G(k) \\ r_1, r_2 = G(k') \end{array} \right\}$$

Claim: G' is a PRG

G' is a PRG means:

$$\left\{ G'(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\} \approx \left\{ r \;\middle|\; r \leftarrow \{0,1\}^{3\lambda} \right\}$$

$$\left\{ G'(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\}$$

$$\left\{ G'(k) \;\middle|\; k \leftarrow \{0,1\}^\lambda \right\} \equiv \left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} k \leftarrow \{0,1\}^\lambda \\ r_0, k' = G(k) \\ r_1, r_2 = G(k') \end{array} \right\}$$

By definition of G'

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} k \leftarrow \{0,1\}^{\lambda} \\ r_0, k' = G(k) \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} k \leftarrow \{0,1\}^\lambda \\ r_0, k' = G(k) \\ r_1, r_2 = G(k') \end{array} \right\}$$

$\equiv$ Refactoring

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^\lambda \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^\lambda \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\approx$$ By PRG security of G

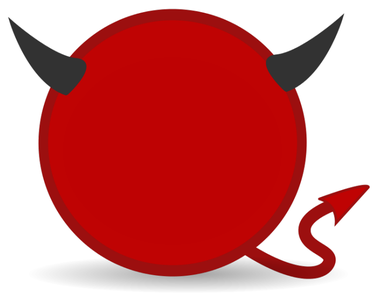
$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^\lambda \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$\approx$ By PRG security of G

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

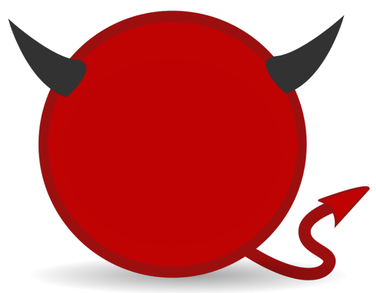
Why does this work?



$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

Adversary A

$\approx$ By PRG security of G



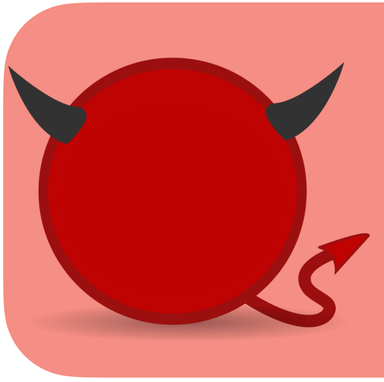
$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

Why does this work?

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \left\{ G(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\} \\ r_1, r_2 = G(k') \end{array} \right\}$$

Adversary B

$\approx$ By PRG security of G



$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

Why does this work?

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$\equiv$ Refactoring

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0 \leftarrow \{0,1\}^{\lambda} \\ r_1, r_2 \leftarrow \left\{ G(k') \;\middle|\; k' \leftarrow \{0,1\}^{\lambda} \right\} \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0 \leftarrow \{0,1\}^\lambda \\ r_1, r_2 \leftarrow \left\{ G(k') \;\middle|\; k' \leftarrow \{0,1\}^\lambda \right\} \end{array} \right\}$$

$$\approx$$ By PRG security of G

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0 \leftarrow \{0,1\}^\lambda \\ r_1, r_2 \leftarrow \{0,1\}^{2\lambda} \end{array} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0 \leftarrow \{0,1\}^{\lambda} \\ r_1, r_2 \leftarrow \{0,1\}^{2\lambda} \end{array} \right\}$$

$$\equiv$$ Refactoring

$$\left\{ r \;\middle|\; r \leftarrow \{0,1\}^{3\lambda} \right\}$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0 \leftarrow \{0,1\}^{\lambda} \\ r_1, r_2 \leftarrow \{0,1\}^{2\lambda} \end{array} \right\}$$

$\equiv$ Refactoring

$$\left\{ r \;\middle|\; r \leftarrow \{0,1\}^{3\lambda} \right\}$$

$$\left\{ G'(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\}$$

$$\approx$$

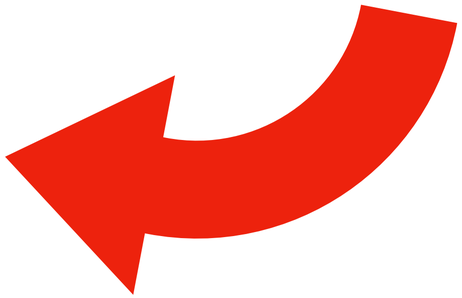
$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\}$$

$$\approx$$

$$\left\{ r \;\middle|\; r \leftarrow \{0,1\}^{3\lambda} \right\}$$

This ensemble is not part of the security definition

It is a ***hybrid experiment***



$$\left\{ G'(k) \;\middle|\; k \leftarrow \{0,1\}^{\lambda} \right\}$$

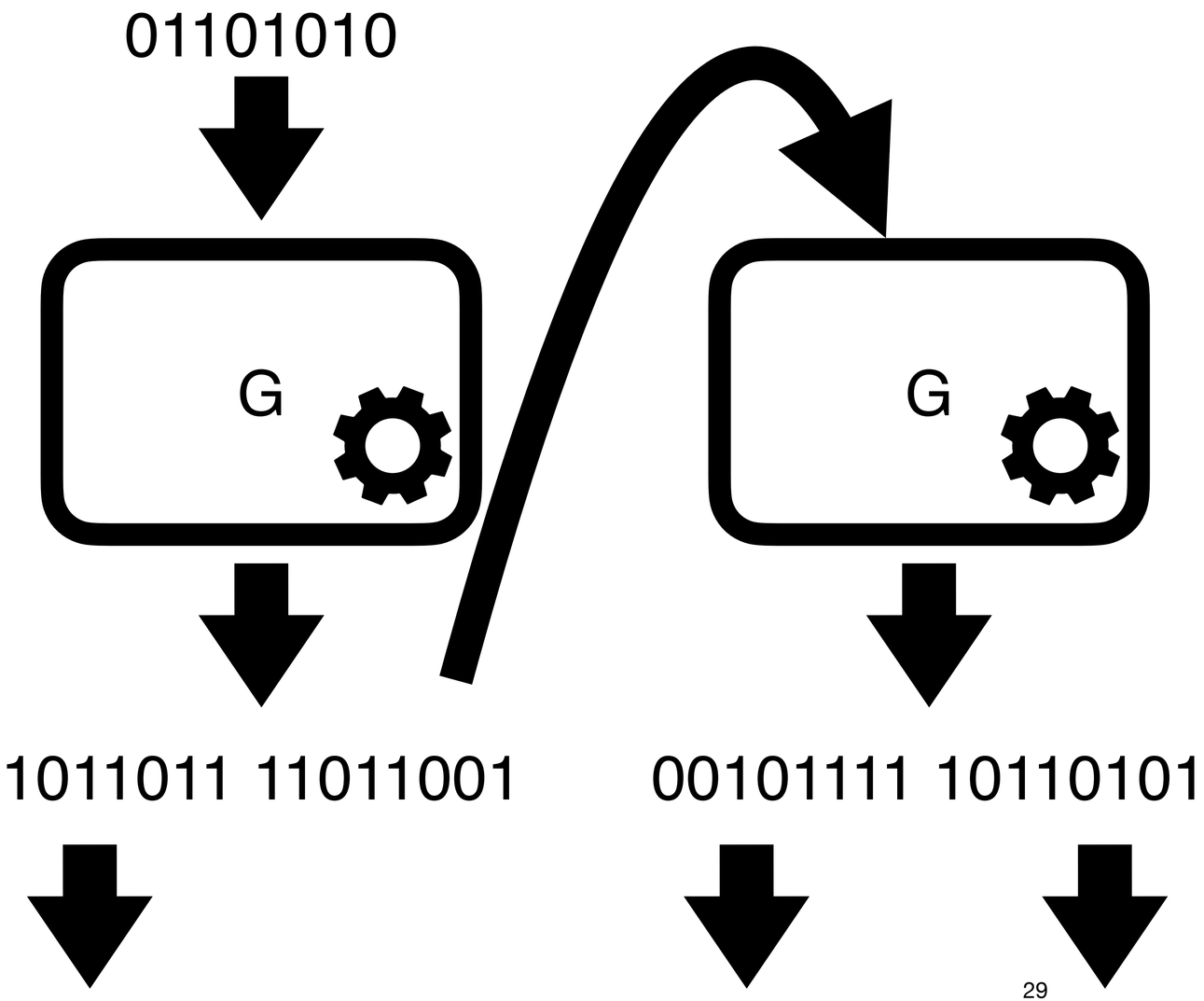
$$\approx$$

$$\left\{ r_0, r_1, r_2 \;\middle|\; \begin{array}{l} r_0, k' \leftarrow \{0,1\}^{2\lambda} \\ r_1, r_2 = G(k') \end{array} \right\} \qquad X \approx Y \text{ and } Y \approx Z \implies X \approx Z$$

$$\approx$$

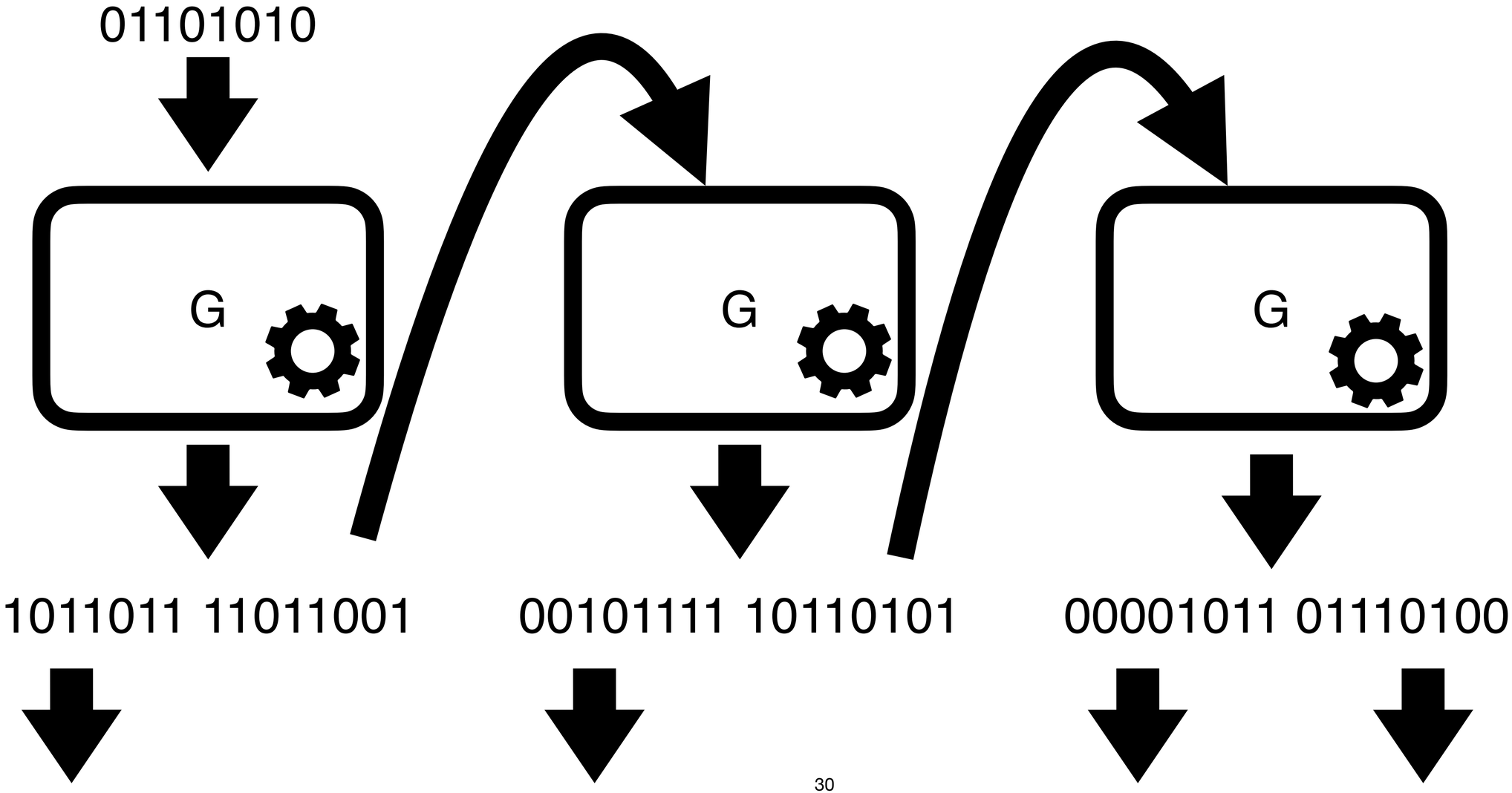
$$\left\{ r \;\middle|\; r \leftarrow \{0,1\}^{3\lambda} \right\}$$

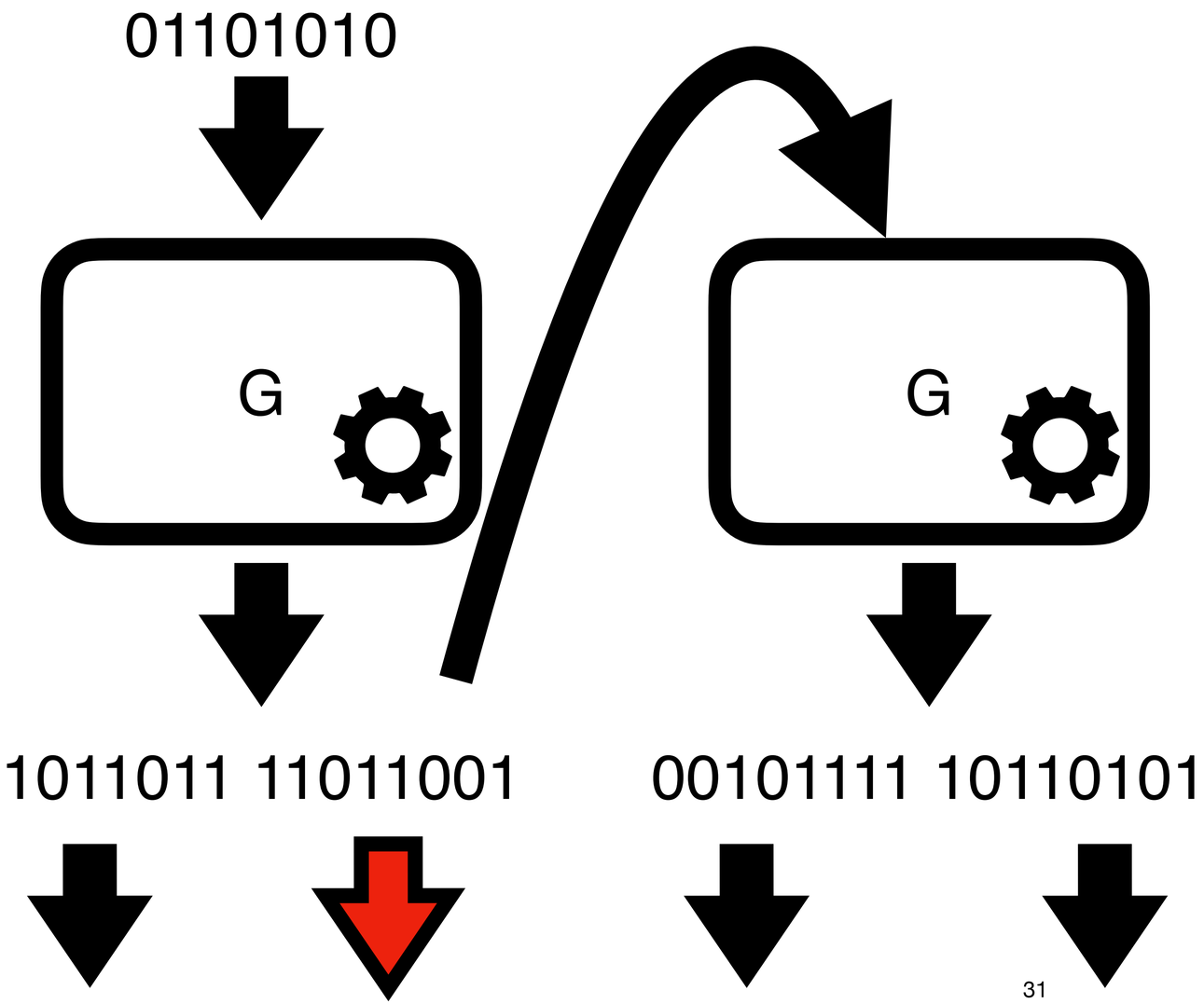
Stretching the output of a PRG



This is a secure PRG

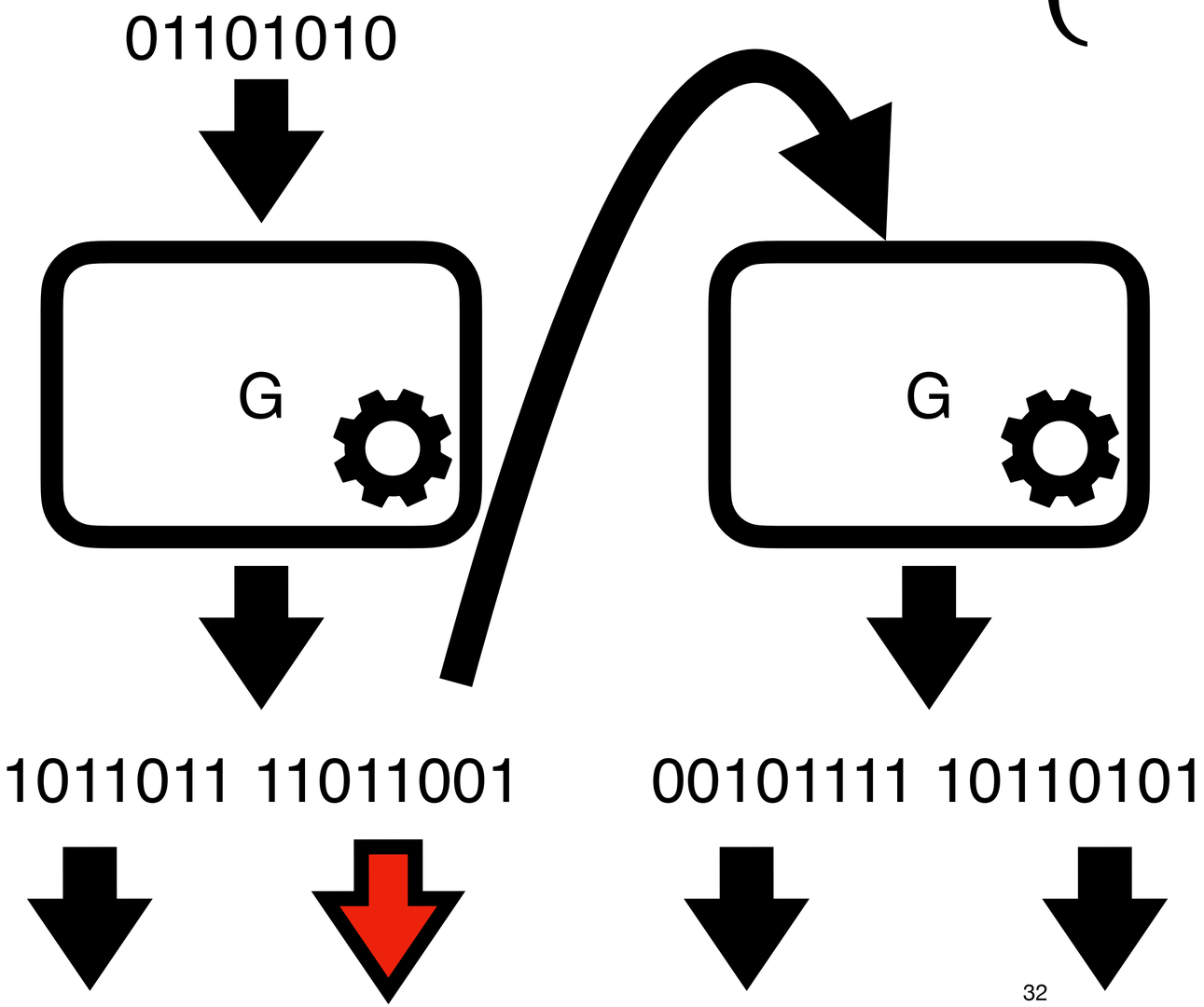
Repeatable any polynomial number of times





Is this a PRG?

$$G'(k) = \left\{ r_0, r_1, r_2, r_3 \;\middle|\; \begin{array}{l} r_0, r_1 = G(k) \\ r_2, r_3 = G(r_1) \end{array} \right\}$$



Is this a PRG?

Today’s objectives

Construct PRG with long stretch from one with short stretch

Introduce the notion of a **security reduction**

Understand an attack