The Provable Security Methodology

Pooya Farshim

Queen's University Belfast/École Normale Supérieure

Mathematical Aspects of Computer Science – Foundations of Cryptography

متدولوژیِ امنیت قابل اثبات

پویا فرشیم

دانشگاه کوئینز بلفاست – اکول نرمال سوپریور

جنبه‌های ریاضی علوم کامپیوتری – مبانی رمز

Unreadable Effectiveness of Mathematics

...the enormous usefulness of mathematics in the natural sciences is something bordering on the mysterious and there is no rational explanation for it.

...دامنه استفاده از ریاضیات در علوم طبیعی بقدری وسیع است که به مرزهای رازآمیز میرسد و هیچگونه توضیح منطقی برای آن وجود ندارد.

1 E.P. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Com. in Pure and Applied Mathematics*, 1960.

Unreadable Effectiveness of Mathematics

> ...the enormous usefulness of mathematics in the natural sciences is something bordering on the mysterious and there is no rational explanation for it.

...دامنه استفاده از ریاضیات در علوم طبیعی بقدری وسیع است که به مرزهای رازآمیز میرسد و هیچگونه توضیح منطقی برای آن وجود ندارد.

1 E.P. Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Com. in Pure and Applied Mathematics*, 1960.

2 R.W. Hamming. The unreasonable effectiveness of mathematics. *The American Mathematical Monthly*, 1980.

Cryptography, *n*

The Oxford English Dictionary defines:

> **cryptography**, *n.* **1.** The art or practice of writing in code or cipher;

Ad Hoc Protocol Deign

Problem

Proposed Solution

Bug!

Revised Solution

...

Implementation

Bug!

...

Source: Bellare & Rogaway. Introduction to Modern Cryptography. 2005.

Cryptography, *n*

The entry continues

> the science of encryption; the branch of cryptology concerned with this (cf. cryptanalysis *n.*). More generally: the study of codes and ciphers; cryptology.

Cryptography, *n*

The entry continues

> the science of encryption; the branch of cryptology concerned with this (cf. cryptanalysis *n.*). More generally: the study of codes and ciphers; cryptology.

A broader definition that's perhaps better suited for us:

> **cryptology**: the scientific/mathematical study of systems in the presence of adversarial behavior.

Provable Security

The seminal work of Goldwasser and Micali [STOC'82] initiated the

Definition-Theorem-Proof approach to design and analysis of protocols.

This consists of:

Provable Security

The seminal work of Goldwasser and Micali [STOC'82] initiated the

Definition-Theorem-Proof approach to design and analysis of protocols.

This consists of:

(1) Formal syntax and correctness of a protocol

Provable Security

The seminal work of Goldwasser and Micali [STOC'82] initiated the

Definition-Theorem-Proof approach to design and analysis of protocols.

This consists of:

(1) Formal syntax and correctness of a protocol

(2) Formulation of a precise definition of security

Provable Security

The seminal work of Goldwasser and Micali [STOC'82] initiated the

Definition-Theorem-Proof approach to design and analysis of protocols.

This consists of:

(1) Formal syntax and correctness of a protocol
(2) Formulation of a precise definition of security
(3) Statement of assumptions

Provable Security

The seminal work of Goldwasser and Micali [STOC'82] initiated the

Definition-Theorem-Proof approach to design and analysis of protocols.

This consists of:

(1) Formal syntax and correctness of a protocol
(2) Formulation of a precise definition of security
(3) Statement of assumptions
(4) Proof that an instance of (1) is secure wrt. definition (2) relative to assumptions (3).

See "Post-Modern Cryptography" [Gol06] for a (vehement!) defense.

The Role of Definitions

- Mathematical treatment

The Role of Definitions

- Mathematical treatment
- Abstraction boundaries

The Role of Definitions

- Mathematical treatment
- Abstraction boundaries
- Direct our design goals

The Role of Definitions

- Mathematical treatment
- Abstraction boundaries
- Direct our design goals
- Comparison of protocols

The Role of Definitions

- Mathematical treatment
- Abstraction boundaries
- Direct our design goals
- Comparison of protocols
- Make it easier to come up with attacks

As Rogaway [Rog04] writes in "On the Role of Definitions in and Beyond Cryptography":

> Definitions shape and direct the way we think.

The Role of Definitions

- Mathematical treatment
- Abstraction boundaries
- Direct our design goals
- Comparison of protocols
- Make it easier to come up with attacks

As Rogaway [Rog04] writes in "On the Role of Definitions in and Beyond Cryptography":

> Definitions shape and direct the way we think. ⚠

(1) Syntax & Correctness

Cannot talk about what an object is supposed to do until the object is defined.

(1) Syntax & Correctness

Cannot talk about what an object is supposed to do
until the object is defined.

Example: What's a one-way function?

$$f : \mathsf{D} \longrightarrow \mathsf{R}$$

(1) Syntax & Correctness

Cannot talk about what an object is supposed to do
until the object is defined.

Example: What's a one-way function?

$$f : \mathrm{D} \longrightarrow \mathrm{R}$$

Syntax first! A (trapdoor) function family is a tuple of poly-time TMs:

(1) Syntax & Correctness

Cannot talk about what an object is supposed to do
until the object is defined.

Example: What's a one-way function?

$$f : \mathsf{D} \longrightarrow \mathsf{R}$$

Syntax first! A (trapdoor) function family is a tuple of poly-time TMs:

1. $\underline{\mathsf{Gen}(1^\lambda)}$: is randomized and outputs (fk, td).
2. $\underline{\mathsf{F}(fk, x)}$: is deterministic; given fk and $x \in \mathsf{D}_{fk}$ outputs $y \in \mathsf{R}_{fk}$.
3. $\underline{\mathsf{F}^{-1}(td, y)}$: is deterministic; given td and $y \in \mathsf{R}_{fk}$ outputs $x \in \mathsf{D}_{fk}$.

(1) Syntax & Correctness

Cannot talk about what an object is supposed to do
until the object is defined.

Example: What's a one-way function?

$$f : \mathsf{D} \longrightarrow \mathsf{R}$$

Syntax first! A (trapdoor) function family is a tuple of poly-time TMs:

1. $\underline{\mathsf{Gen}(1^\lambda)}$: is randomized and outputs (fk, td).
2. $\underline{\mathsf{F}(fk, x)}$: is deterministic; given fk and $x \in \mathsf{D}_{fk}$ outputs $y \in \mathsf{R}_{fk}$.
3. $\underline{\mathsf{F}^{-1}(td, y)}$: is deterministic; given td and $y \in \mathsf{R}_{fk}$ outputs $x \in \mathsf{D}_{fk}$.
4. $\underline{\mathsf{D}(fk)}$: is randomized and given fk returns an element of D_{fk}.

(1) Syntax & Correctness

Correctness: For all choices of inputs (in appropriate spaces):

$$\mathsf{F}^{-1}(td, \mathsf{F}(fk, x)) = x \; .$$

(1) Syntax & Correctness

Correctness: For all choices of inputs (in appropriate spaces):

$$\mathsf{F}^{-1}(td, \mathsf{F}(fk, x)) = x \ .$$

Or demand $\mathsf{F}^{-1}(td, \cdot)$ **samples** a random pre-image.

(2) Security

Ingredients:

- What it means to break a system.

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

    Recover keys,

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

    Recover keys, plaintexts,

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

  Recover keys, plaintexts, or **some** information about plaintexts?

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

  Recover keys, plaintexts, or **some** information about plaintexts?

- What the adversary's resources are.

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

  Recover keys, plaintexts, or **some** information about plaintexts?

- What the adversary's resources are.

  Does it see a single or multiple ciphertexts?

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

  Recover keys, plaintexts, or **some** information about plaintexts?

- What the adversary's resources are.

  Does it see a single or multiple ciphertexts? For how long can it run?

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

  Recover keys, plaintexts, or **some** information about plaintexts?

- What the adversary's resources are.

  Does it see a single or multiple ciphertexts? For how long can it run?

- A metric to measure adversary's success/advantage.

(2) Security

Ingredients:

- What it means to break a system. Given ciphertexts

  Recover keys, plaintexts, or **some** information about plaintexts?

- What the adversary's resources are.

  Does it see a single or multiple ciphertexts? For how long can it run?

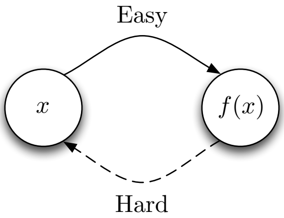
- A metric to measure adversary's success/advantage.

A cryptosystem is called secure if

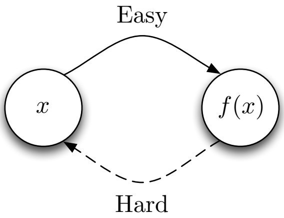
Any adversary of specified resources has small advantage (according to the metric) in achieving the specified break.

Example: One-Way Functions

Example: One-Way Functions



Example: One-Way Functions



Formalization involves multiple choices:

- Computing **a** pre-image vs. **the** pre-image: Is $x \mapsto 0$ useful?
- How is $f(x)$ sampled? Is it sampled at all?
- Some pre-image can always be found in exponential time.
- And can always guess x with nonzero probability.
- How "small" a probability do we want?

Negligible Functions

A negligible function $\mu : \mathbb{N} \longrightarrow \mathbb{R}$ approaches zero faster that $1/\mathrm{poly}(\lambda)$ for any $\mathrm{poly}(\lambda)$.[1] In other terms

$$\mu(\lambda) \in \lambda^{-\omega(1)} \ .$$

[1] Formally, $(\forall d \in \mathbb{N})(\exists \lambda_d \in \mathbb{N})(\forall \lambda \geq \lambda_d) \ : \ |\mu(\lambda)| \leq 1/\lambda^d$.

Negligible Functions

A negligible function $\mu : \mathbb{N} \longrightarrow \mathbb{R}$ approaches zero faster that $1/\mathrm{poly}(\lambda)$ for any $\mathrm{poly}(\lambda)$.[1] In other terms

$$\mu(\lambda) \in \lambda^{-\omega(1)} \ .$$

Examples:

$$2^{-\lambda} \quad \text{and} \quad 2^{-\lambda^{\epsilon}} \ .$$

[1] Formally, $(\forall d \in \mathbb{N})(\exists \lambda_d \in \mathbb{N})(\forall \lambda \geq \lambda_d) \ : \ |\mu(\lambda)| \leq 1/\lambda^d$.

Negligible Functions

A negligible function $\mu : \mathbb{N} \longrightarrow \mathbb{R}$ approaches zero faster that $1/\mathrm{poly}(\lambda)$ for any $\mathrm{poly}(\lambda)$.[1] In other terms

$$\mu(\lambda) \in \lambda^{-\omega(1)} \ .$$

Examples:

$$2^{-\lambda} \quad \text{and} \quad 2^{-\lambda^{\epsilon}} \ .$$

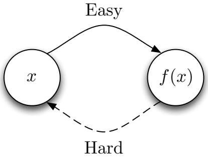
Define

$$\mathrm{Negl} := \{ \text{ all negligible } \mu : \mathbb{N} \longrightarrow \mathbb{R} \ \}.$$

[1] Formally, $(\forall d \in \mathbb{N})(\exists \lambda_d \in \mathbb{N})(\forall \lambda \geq \lambda_d) \ : \ |\mu(\lambda)| \leq 1/\lambda^{d}$.

Example: One-Way Functions

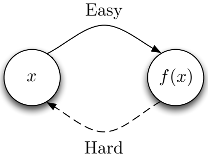
Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):



$\underline{\mathrm{OW}^{\mathcal{A}}_{\mathsf{FF}}(1^{\lambda})}$:

Example: One-Way Functions

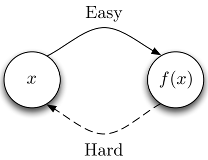
Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):



$$\underline{\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)}:$$
$$(fk, td) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$$

Example: One-Way Functions

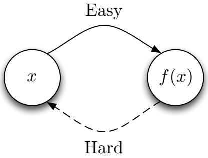
Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):



$$\underline{\mathrm{OW}^{A}_{\mathsf{FF}}(1^{\lambda})}:$$
$$(fk, td) \leftarrow_{\$} \mathsf{Gen}(1^{\lambda})$$
$$x \leftarrow_{\$} \mathsf{D}(fk)$$
$$y \leftarrow \mathsf{F}(fk, x)$$

Example: One-Way Functions

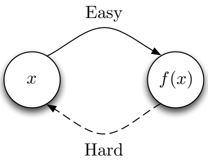
Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):



$\underline{\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)}$:
$(fk, td) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$x \leftarrow_{\$} \mathsf{D}(fk)$
$y \leftarrow \mathsf{F}(fk, x)$
$\bar{x} \leftarrow_{\$} A(fk, y)$

Example: One-Way Functions

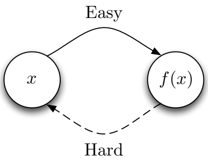
Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):



$\underline{\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)}$:
$(fk, td) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$
$x \leftarrow_\$ \mathsf{D}(fk)$
$y \leftarrow \mathsf{F}(fk, x)$
$\bar{x} \leftarrow_\$ A(fk, y)$
$\bar{y} \leftarrow \mathsf{F}(fk, \bar{x})$
return $(\bar{y} = y)$

Example: One-Way Functions

Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):

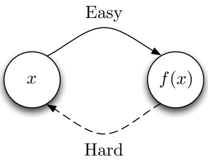


$\underline{\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)}$:
$(fk, td) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$
$x \leftarrow_\$ \mathsf{D}(fk)$
$y \leftarrow \mathsf{F}(fk, x)$
$\bar{x} \leftarrow_\$ A(fk, y)$
$\bar{y} \leftarrow \mathsf{F}(fk, \bar{x})$
return $(\bar{y} = y)$

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) := \Pr[\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)]$$

Example: One-Way Functions

Security: Hard to invert $y := \mathsf{F}(fk, x)$ for random (fk, x):



$\underline{\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)}$:
$(fk, td) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$
$x \leftarrow_{\$} \mathsf{D}(fk)$
$y \leftarrow \mathsf{F}(fk, x)$
$\bar{x} \leftarrow_{\$} A(fk, y)$
$\bar{y} \leftarrow \mathsf{F}(fk, \bar{x})$
return $(\bar{y} = y)$

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) := \Pr[\mathrm{OW}^{A}_{\mathsf{FF}}(1^\lambda)]$$

Require:

$$\forall \text{ ppt TMs } A \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \in \mathrm{Negl}\ .$$

Why Look at OWF?

Our ultimate goal:

Build highly secure and efficient cryptosystems.

Why Look at OWF?

Our ultimate goal:

Build highly secure and efficient cryptosystems.

But most crypto implies $\mathbf{P} \neq \mathbf{NP}$.

Why Look at OWF?

Our ultimate goal:

Build highly secure and efficient cryptosystems.

But most crypto implies $\mathbf{P} \neq \mathbf{NP}$. This raises the question:

Can we base crypto on complexity assumptions
that are as weak and as general as possible?

Why Look at OWF?

Our ultimate goal:

Build highly secure and efficient cryptosystems.

But most crypto implies $\mathbf{P} \neq \mathbf{NP}$. This raises the question:

Can we base crypto on complexity assumptions
that are as weak and as general as possible?

As we shall see: “OWFs exist” is a very safe assumption.

The "Right" Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

The “Right” Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].

The “Right” Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

  Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].

- Worst-case OWFs: For all non-uniform pt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \neq 1 \ .$$

The "Right" Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].

- Worst-case OWFs: For all non-uniform pt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \neq 1 \ .$$

I.e., any A fails ∞ly often.

The "Right" Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

  Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].

- Worst-case OWFs: For all non-uniform pt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \neq 1 \ .$$

  I.e., any A fails ∞ly often.

- Point-wise vs. uniform bound:

$$\forall A \, \exists \mu \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$
$$\exists \mu \, \forall A \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$

The “Right” Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

  Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].

- Worst-case OWFs: For all non-uniform pt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \neq 1 \ .$$

  I.e., any A fails ∞ly often.

- Point-wise vs. uniform bound:

$$\forall A \, \exists \mu \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$

$$\exists \mu \, \forall A \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$

  Equivalent for ppt A and negligible μ [Bel97].

The "Right" Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda)\ .$$

  Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].

- Worst-case OWFs: For all non-uniform pt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \neq 1\ .$$

  I.e., any A fails ∞ly often.

- Point-wise vs. uniform bound:

$$\forall A\, \exists \mu\ :\ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$
$$\exists \mu\, \forall A\ :\ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$

  Equivalent for ppt A and negligible μ [Bel97].

- Sub-exponentially large runtime 2^{λ^δ} and/or small advantage $2^{-\lambda^\epsilon}$
- Quantum/non-uniformity

The "Right" Notion?

- Weak OWFs: There is a $\mathrm{poly}(\lambda)$ such that for all ppt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq 1 - 1/\mathrm{poly}(\lambda) \ .$$

  Turns out $\exists$ weak OWF $\iff$ and $\exists$ strong OWF [Gol01].
- Worst-case OWFs: For all non-uniform pt A:

$$\mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \neq 1 \ .$$

  I.e., any A fails ∞ly often.
- Point-wise vs. uniform bound:

$$\forall A \, \exists \mu \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$
$$\exists \mu \, \forall A \ : \ \mathbf{Adv}^{\mathrm{ow}}_{\mathsf{FF},A}(\lambda) \leq \mu(\lambda)$$

  Equivalent for ppt A and negligible μ [Bel97].
- Sub-exponentially large runtime $2^{\lambda^{\delta}}$ and/or small advantage $2^{-\lambda^{\epsilon}}$
- Quantum/non-uniformity

Can we relate these notions?

Formulating Definitions

- Rooted in (our inherent?) mathematical intuition

Formulating Definitions

- Rooted in (our inherent?) mathematical intuition
- Convenience and elegance

Formulating Definitions

- Rooted in (our inherent?) mathematical intuition
- Convenience and elegance
- Relations with other notions

Formulating Definitions

- Rooted in (our inherent?) mathematical intuition
- Convenience and elegance
- Relations with other notions
- As a security goal: strength and ability to model real-world attacks

Formulating Definitions

- Rooted in (our inherent?) mathematical intuition
- Convenience and elegance
- Relations with other notions
- As a security goal: strength and ability to model real-world attacks
- As **assumptions**: safety
  - Failure of cryptanalytic attacks
  - Simplicity or clarity
  - Generic vs. specific, multiple candidates, universality
  - Weakness: implication by other assumptions
  - Structure: falsifiability, simplicity, average-case to worse-case reduction, etc.

Formulating Definitions

- Rooted in (our inherent?) mathematical intuition
- Convenience and elegance
- Relations with other notions
- As a security goal: strength and ability to model real-world attacks
- As **assumptions**: safety
  - Failure of cryptanalytic attacks
  - Simplicity or clarity
  - Generic vs. specific, multiple candidates, universality
  - Weakness: implication by other assumptions
  - Structure: falsifiability, simplicity, average-case to worse-case reduction, etc.
- Ultimately: How accepting the community is, and how fruitful a definition turns out to be.

(3) Assumptions [GK15]

Definition (Crypto Assumption)

A (search) cryptographic assumption is an interactive protocol

$$\langle \mathsf{C}(1^\lambda), A(1^\lambda)\rangle$$

between a challenger C and an adversary A that terminates with C outputting T/F. The assumption holds in the average case if for all ppt A

$$\mathbf{Adv}_{\mathsf{C},A}(\lambda) := \Pr[\langle \mathsf{C}(1^\lambda), A(1^\lambda)\rangle] \in \mathrm{Negl}\ .$$

It holds in the worst case if for all non-uniform pt A

$$\mathbf{Adv}_{\mathsf{C},A}(\lambda) \neq 1\ .$$

When C is ppt, we call the assumption **falsifiable** [Nao03].

Simple(r) Assumptions

Definition (Search complexity assumption)

Protocol $\langle \mathsf{C}, A \rangle$ for some ppt (D, R) can be written as:

- $x \leftarrow \mathsf{D}(1^\lambda; r)$
- $y \leftarrow_{\$} A(x)$
- if “public” then $r \leftarrow \varepsilon$
- return $\mathsf{R}(x, y, r)$

Simple(r) Assumptions

Definition (Search complexity assumption)

Protocol $\langle \mathsf{C}, A \rangle$ for some ppt (D, R) can be written as:

- $x \leftarrow \mathsf{D}(1^\lambda; r)$
- $y \leftarrow_\$ A(x)$
- if "public" then $r \leftarrow \varepsilon$
- return $\mathsf{R}(x, y, r)$

Public Examples: Factoring: $(p, q) \mapsto pq$, Discrete Logs etc.
Private Example: CDH: Given (g^x, g^y) find g^{xy}.

Simple(r) Assumptions

Definition (Search complexity assumption)

Protocol $\langle \mathsf{C}, A \rangle$ for some ppt (D, R) can be written as:

- $x \leftarrow \mathsf{D}(1^\lambda; r)$
- $y \leftarrow_{\$} A(x)$
- if "public" then $r \leftarrow \varepsilon$
- return $\mathsf{R}(x, y, r)$

Public Examples: Factoring: $(p, q) \mapsto pq$, Discrete Logs etc.
Private Example: CDH: Given (g^x, g^y) find g^{xy}.

To what extent are these existentially equivalent?

Average-Case and Worst-Case

Definition

Assumption C has an average-case to worst-case reduction if for all ppt A there is a non-uniform pt B such that

$$\mathbf{Adv}_{\mathsf{C},A}(\lambda) \notin \mathrm{Negl} \implies \mathbf{Adv}_{\mathsf{C},B}(\lambda) = 1 \ .$$

Average-Case and Worst-Case

Definition

Assumption C has an average-case to worst-case reduction if for all ppt A there is a non-uniform pt B such that

$$\mathbf{Adv}_{\mathsf{C},A}(\lambda) \notin \mathrm{Negl} \implies \mathbf{Adv}_{\mathsf{C},B}(\lambda) = 1 \ .$$

Example: Discrete Logs: Invert $(g, x) \mapsto (g, g^x)$.

- Randomize (g, g^x) as (g^r, g^{xs}) for random (r, s).
- Given y, return $\frac{ry}{s}$.

For OWFs: Related to basing OWFs on $\mathbf{NP}$ hardness.

Average-Case and Worst-Case

Definition

Assumption C has an average-case to worst-case reduction if for all ppt A there is a non-uniform pt B such that

$$\mathbf{Adv}_{\mathsf{C},A}(\lambda) \notin \mathrm{Negl} \implies \mathbf{Adv}_{\mathsf{C},B}(\lambda) = 1 \ .$$

Example: Discrete Logs: Invert $(g, x) \mapsto (g, g^x)$.

- Randomize (g, g^x) as (g^r, g^{xs}) for random (r, s).
- Given y, return $\frac{ry}{s}$.

For OWFs: Related to basing OWFs on $\mathbf{NP}$ hardness.

What about other primitives?

Can we base average-case hash functions on worst-case hash functions?

Universality of OWF

What's the difference between:

> "Assume OWFs exist."

Universality of OWF

What's the difference between:

"Assume OWFs exist." and "Assume $(p, q) \mapsto pq$ is one-way."

Universality of OWF

What's the difference between:

"Assume OWFs exist." and "Assume $(p, q) \mapsto pq$ is one-way."

"Generic hardness."

Universality of OWF

What's the difference between:

"Assume OWFs exist." and "Assume $(p, q) \mapsto pq$ is one-way."

"Generic hardness." vs. "Concrete/specific hardness."

Universality of OWF

What's the difference between:

"Assume OWFs exist." and "Assume $(p, q) \mapsto pq$ is one-way."

"Generic hardness." vs. "Concrete/specific hardness."

Not even clear if the first is falsifiable!

Universality of OWF

What's the difference between:

"Assume OWFs exist." and "Assume $(p, q) \mapsto pq$ is one-way."

"Generic hardness." vs. "Concrete/specific hardness."

Not even clear if the first is falsifiable! But

Theorem (Universal OWF [Lev87])

There is a function family that is one-way if and only if OWFs exist.

Universality of OWF

What's the difference between:

"Assume OWFs exist." and "Assume $(p, q) \mapsto pq$ is one-way."

"Generic hardness." vs. "Concrete/specific hardness."

Not even clear if the first is falsifiable! But

Theorem (Universal OWF [Lev87])

There is a function family that is one-way if and only if OWFs exist.

Formalize universality. Which primitives are universal?

(4) Security Results

Results in provable security often look like this:

Theorem

If scheme D is secure wrt. its definition of security, then construction C is secure wrt. its definition of security. More precisely, for any $t(\lambda)$-time A, there is a $t'(\lambda)$-time adversary B such that:

$$\mathbf{Adv}^{\mathrm{sec1}}_{C,A}(\lambda) \leq \tau(\lambda) \cdot \mathbf{Adv}^{\mathrm{sec2}}_{D,B}(\lambda) + \mu(\lambda)$$

for a (polynomial) function $\tau(\lambda)$ and a negligible function $\mu(\lambda)$.

(4) Security Results

Results in provable security often look like this:

Theorem

If scheme D is secure wrt. its definition of security, then construction C is secure wrt. its definition of security. More precisely, for any $t(\lambda)$-time A, there is a $t'(\lambda)$-time adversary B such that:

$$\mathbf{Adv}^{\mathrm{sec1}}_{C,A}(\lambda) \le \tau(\lambda) \cdot \mathbf{Adv}^{\mathrm{sec2}}_{D,B}(\lambda) + \mu(\lambda)$$

for a (polynomial) function $\tau(\lambda)$ and a negligible function $\mu(\lambda)$.

A proof is **tight** if

$$t(\lambda) \approx t'(\lambda) \quad \text{and} \quad \tau(\lambda) \approx 1 \quad \text{and} \quad \mu(\lambda) \approx 0 \ .$$

This means most of the security inherent in D is **preserved**.

(4) Security Results

Results in provable security often look like this:

Theorem

If scheme D is secure wrt. its definition of security, then construction C is secure wrt. its definition of security. More precisely, for any $t(\lambda)$-time A, there is a $t'(\lambda)$-time adversary B such that:

$$\mathbf{Adv}^{\mathrm{sec1}}_{C,A}(\lambda) \leq \tau(\lambda) \cdot \mathbf{Adv}^{\mathrm{sec2}}_{D,B}(\lambda) + \mu(\lambda)$$

for a (polynomial) function $\tau(\lambda)$ and a negligible function $\mu(\lambda)$.

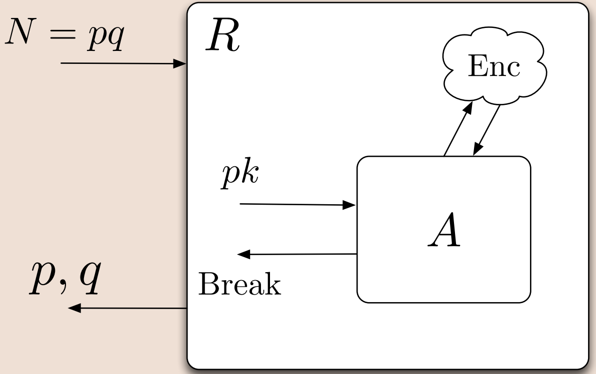
A proof is **tight** if

$$t(\lambda) \approx t'(\lambda) \quad \text{and} \quad \tau(\lambda) \approx 1 \quad \text{and} \quad \mu(\lambda) \approx 0 \ .$$

This means most of the security inherent in D is **preserved**.
Guides us how to set key sizes.

Proofs via Reductions

Proof.



□

Significance of Proofs

Significance of Proofs

- Lots of protocols are broken ...
- Yet AES, Factoring, etc. have not been broken.

Significance of Proofs

- Lots of protocols are broken ...
- Yet AES, Factoring, etc. have not been broken.
- Proofs allow protocols to **inherit** this strength.

Significance of Proofs

- Lots of protocols are broken ...
- Yet AES, Factoring, etc. have not been broken.
- Proofs allow protocols to **inherit** this strength.
- And tell us **how much** strength is inherited.

Significance of Proofs

- Lots of protocols are broken ...
- Yet AES, Factoring, etc. have not been broken.
- Proofs allow protocols to **inherit** this strength.
- And tell us **how much** strength is inherited.

Nowadays provable security is required to support standardization.
(e.g., by ISO, IETF, CAESAR, etc.)

Significance of Proofs

- Lots of protocols are broken ...
- Yet AES, Factoring, etc. have not been broken.
- Proofs allow protocols to **inherit** this strength.
- And tell us **how much** strength is inherited.

Nowadays provable security is required to support standardization.
(e.g., by ISO, IETF, CAESAR, etc.)

Caveats:

- Provable security never proves security in an absolute sense.
- Security is always relative to the assumptions and the model.
- Definitions, assumptions, protocols and proofs change!

Collision Resistance

Security: Hard to find $x_1 \neq x_2$ such that $F(fk, x_1) = F(fk, x_2)$.

Collision Resistance

Security: Hard to find $x_1 \neq x_2$ such that $F(fk, x_1) = F(fk, x_2)$.

$$\underline{\mathrm{CR}^{A}_{\mathsf{FF}}(1^\lambda)}:$$
$$(fk, td) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$$

Collision Resistance

Security: Hard to find $x_1 \neq x_2$ such that $F(fk, x_1) = F(fk, x_2)$.

$$\underline{\mathrm{CR}^{A}_{\mathsf{FF}}(1^\lambda)}:$$
$$(fk, td) \leftarrow_{\$} \mathsf{Gen}(1^\lambda)$$
$$(x_1, x_2) \leftarrow_{\$} A(fk)$$

Collision Resistance

Security: Hard to find $x_1 \neq x_2$ such that $F(fk, x_1) = F(fk, x_2)$.

$$\begin{array}{|l|}\hline \underline{\mathrm{CR}^{A}_{\mathsf{FF}}(1^\lambda)}: \\ (fk, td) \leftarrow_{\$} \mathsf{Gen}(1^\lambda) \\ (x_1, x_2) \leftarrow_{\$} A(fk) \\ y_1 \leftarrow \mathsf{F}(fk, x_1) \\ y_2 \leftarrow \mathsf{F}(fk, x_2) \\ \text{return } (x_1 \neq x_2 \wedge y_1 = y_2) \\ \hline \end{array}$$

Collision Resistance

Security: Hard to find $x_1 \neq x_2$ such that $F(fk, x_1) = F(fk, x_2)$.

$$\begin{array}{|l|}\hline \underline{\mathrm{CR}^{A}_{\mathsf{FF}}(1^\lambda)}: \\ (fk, td) \leftarrow_{\$} \mathsf{Gen}(1^\lambda) \\ (x_1, x_2) \leftarrow_{\$} A(fk) \\ y_1 \leftarrow \mathsf{F}(fk, x_1) \\ y_2 \leftarrow \mathsf{F}(fk, x_2) \\ \text{return } (x_1 \neq x_2 \wedge y_1 = y_2) \\ \hline \end{array}$$

Require:

$$\forall \text{ ppt } A \ : \ \mathbf{Adv}^{\mathrm{cr}}_{\mathsf{FF},A}(\lambda) := \Pr[\mathrm{CR}^{A}_{\mathsf{FF}}(1^\lambda)] \in \mathrm{Negl} \ .$$

Relating OWFs and CRFs

Relating OWFs and CRFs

- There are no unkeyed CRFs in the non-uniform model.

Relating OWFs and CRFs

- There are no unkeyed CRFs in the non-uniform model.
- Faster **generic** attacks:

	OWF	CRF
Classical	$\Theta(q/2^\lambda)$	$\Theta(q^2/2^\lambda)$
Quantum	$\Theta(q^2/2^\lambda)$	$\Theta(q^3/2^\lambda)$

Relating OWFs and CRFs

- There are no unkeyed CRFs in the non-uniform model.
- Faster **generic** attacks:

	OWF	CRF
Classical	$\Theta(q/2^\lambda)$	$\Theta(q^2/2^\lambda)$
Quantum	$\Theta(q^2/2^\lambda)$	$\Theta(q^3/2^\lambda)$

- Every compressing collision resistant function is also one-way.

Relating OWFs and CRFs

- There are no unkeyed CRFs in the non-uniform model.
- Faster **generic** attacks:

	OWF	CRF
Classical	$\Theta(q/2^\lambda)$	$\Theta(q^2/2^\lambda)$
Quantum	$\Theta(q^2/2^\lambda)$	$\Theta(q^3/2^\lambda)$

- Every compressing collision resistant function is also one-way. But not every OWF is a CRF.

Relating OWFs and CRFs

- There are no unkeyed CRFs in the non-uniform model.
- Faster **generic** attacks:

	OWF	CRF
Classical	$\Theta(q/2^\lambda)$	$\Theta(q^2/2^\lambda)$
Quantum	$\Theta(q^2/2^\lambda)$	$\Theta(q^3/2^\lambda)$

- Every compressing collision resistant function is also one-way. But not every OWF is a CRF.
- In fact much more is true: There is an oracle relative to which OWFs exist but CRFs don't.

Relating OWFs and CRFs

- There are no unkeyed CRFs in the non-uniform model.
- Faster **generic** attacks:

	OWF	CRF
Classical	$\Theta(q/2^\lambda)$	$\Theta(q^2/2^\lambda)$
Quantum	$\Theta(q^2/2^\lambda)$	$\Theta(q^3/2^\lambda)$

- Every compressing collision resistant function is also one-way. But not every OWF is a CRF.
- In fact much more is true: There is an oracle relative to which OWFs exist but CRFs don't. $\implies$ No fully BB reduction from CRFs to OWFs [Sim98].

Relating OWFs and CRFs

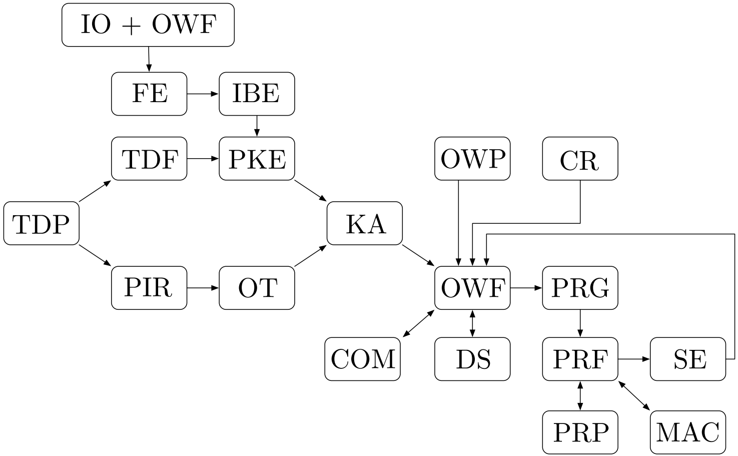
- There are no unkeyed CRFs in the non-uniform model.
- Faster **generic** attacks:

	OWF	CRF
Classical	$\Theta(q/2^\lambda)$	$\Theta(q^2/2^\lambda)$
Quantum	$\Theta(q^2/2^\lambda)$	$\Theta(q^3/2^\lambda)$

- Every compressing collision resistant function is also one-way. But not every OWF is a CRF.
- In fact much more is true: There is an oracle relative to which OWFs exist but CRFs don't. $\implies$ No fully BB reduction from CRFs to OWFs [Sim98].

OWF is (strictly) weaker than CRF as an assumption.

Landscape of Primitives



Thanks!