

Science of Security

Perspectives from Cryptography

Ronald L. Rivest

Viterbi Professor of EECS
MIT, Cambridge, MA

JASON meeting
2010-06-29



Outline

Introduction

Security definitions in terms of games

Example – Semantic Security of PK Encryption

Example – Digital Signatures

Other cryptographic examples

Reusability and Composability

Defining non-crypto security properties



“Can cyber security become a science?”



“Can cyber security become a science?”

=

“Can we give cyber security rigorous foundations, with precise definitions and useful design principles?”



Short summary of this talk:

- ▶ If you want to do “science,” then **precise definitions** are important.

Short summary of this talk:

- ▶ If you want to do “science,” then **precise definitions** are important.
- ▶ Cryptographers have a way of defining security properties that may be more widely useful. (Not unique to cryptography...)



Why is (cyber) security hard?

- ▶ Security hard because of **adversary**



- ▶ Not like earthquakes, floods, natural disasters
- ▶ More like agriculture, health care, sports, war
- ▶ Adversary can be **smart, adaptive, powerful, persistent**

Cryptography

- ▶ Cryptography is “using math to communicate and compute securely in presence of an adversary”
- ▶ math = complexity theory + algebra + probability + $\dots$
- ▶ Cryptography is mature discipline (www.iacr.org)
- ▶ Can successful crypto research paradigms be used more generally in cyber security?



Cryptography

- ▶ Cryptography is “using math to communicate and compute securely in presence of an adversary”
- ▶ math = complexity theory + algebra + probability + ...
- ▶ Cryptography is mature discipline (www.iacr.org)
- ▶ Can successful crypto research paradigms be used more generally in cyber security?
- ▶ Perhaps – cryptographers are particularly good at defining security properties, and in particular defining security in terms of games...



Defining security property in terms of games

- ▶ Goal is to make sure a security property is **well-defined**.
- ▶ Define security property in terms of a **game** for adversary.
- ▶ Game models real-world situation, but may give adversary more (or less) power than he has in reality.
- ▶ Game definition specifies when adversary **wins**.
- ▶ System **satisfies security property** if adversary can't win significantly more often than by playing naively, without breaking one of the assumptions (solving a hard problem, stealing secret key, ...)



Example – Semantic Security of PK Encryption

“When is a public-key encryption method secure?”

– Goldwasser and Micali (1982)



Example – Semantic Security of PK Encryption

- ▶ A PK encryption method is **semantically secure** if no **adversary Adv** can win the following game against a **challenger Ch** with probability noticeably better than $1/2$:



Example – Semantic Security of PK Encryption

- ▶ A PK encryption method is **semantically secure** if no **adversary Adv** can win the following game against a **challenger Ch** with probability noticeably better than $1/2$:

Semantic Security Game:

- ▶ **Ch** makes new public key PK and gives it to **Adv**
- ▶ **Adv** gives **Ch** distinct messages x and y of same length
- ▶ **Ch** randomly picks one of them, call it z
- ▶ **Ch** encrypts z with PK , sends ciphertext to **Adv**
- ▶ **Adv** guesses whether $z = x$ or $z = y$, and wins if correct

Notes and assumptions for this game

- ▶ Notes:
 - ▶ semantically secure encryption must be **randomized!**
 - ▶ **Adv** can easily win with probability exactly $1/2$
- ▶ Assume:
 - ▶ **Adv** can't "solve hard problems"
 - ▶ **Adv** doesn't see **Ch**'s random bits.
 - ▶ **Adv** gets no information about secret key of **Ch** or choice of message except via ciphertext. Side-channels and key-leakage are not modelled here.

Notes and assumptions for this game

- ▶ Notes:
 - ▶ semantically secure encryption must be **randomized!**
 - ▶ **Adv** can easily win with probability exactly $1/2$
- ▶ Assume:
 - ▶ **Adv** can't "solve hard problems"
 - ▶ **Adv** doesn't see **Ch**'s random bits.
 - ▶ **Adv** gets no information about secret key of **Ch** or choice of message except via ciphertext. Side-channels and key-leakage are not modelled here.
- ▶ Relaxing assumptions $\implies$ new games



Notes and assumptions for this game

- ▶ Notes:
 - ▶ semantically secure encryption must be **randomized!**
 - ▶ **Adv** can easily win with probability exactly $1/2$
- ▶ Assume:
 - ▶ **Adv** can't "solve hard problems"
 - ▶ **Adv** doesn't see **Ch**'s random bits.
 - ▶ **Adv** gets no information about secret key of **Ch** or choice of message except via ciphertext. Side-channels and key-leakage are not modelled here.
- ▶ Relaxing assumptions $\implies$ new games
 $\implies$ new security properties



Variations and Extensions

- ▶ **[Chosen ciphertext attack:]** After seeing challenge ciphertext, but before producing his guess, **Adv** may request **Ch** to decrypt any other ciphertext(s).

This new game defines a new security property!

Variations and Extensions

- ▶ **[Chosen ciphertext attack:]** After seeing challenge ciphertext, but before producing his guess, **Adv** may request **Ch** to decrypt any other ciphertext(s).
This new game defines a new security property!
- ▶ **[Key leakage:]** Whenever **Ch** decrypts something, **Adv** learns a certain fraction of **Ch** 's secret key
This new game defines a new security property!

Example – Digital Signature Security

“When is a digital signature method secure?”

Example – Digital Signature Security

A digital signature method is **secure against existential forgery based on an adaptive chosen-message attack** if no adversary **Adv** can win the following game against a challenger **Ch** with more than negligible probability:



Example – Digital Signature Security

A digital signature method is **secure against existential forgery based on an adaptive chosen-message attack** if no adversary **Adv** can win the following game against a challenger **Ch** with more than negligible probability:

Adaptive Existential Forgery Game:

- ▶ **Ch** gives new signature verification key PK to **Adv**
- ▶ **Adv** repeatedly:
 - ▶ creates a new message x and gives it to **Ch**
 - ▶ receives **Ch**'s signature on x
- ▶ The adversary wins if he can produce a nontrivial forgery: a message y that is different than any x , and a z that verifies as a valid signature for y with PK .

Research paradigm

- ▶ **Start** with definition of security mechanism, in terms of its basic functionality and in terms of its security properties, the latter defined in terms of **game(s)**.

Research paradigm

- ▶ **Start** with definition of security mechanism, in terms of its basic functionality and in terms of its security properties, the latter defined in terms of **game(s)**.
- ▶ Then try to see if it is **realizable** (perhaps with additional assumption, and/or the use of previously defined mechanisms – e.g. can you realize a digital signature secure against existential forgery based on adaptive chosen message attack using a public key encryption scheme that is semantically secure?).

Research paradigm

- ▶ **Start** with definition of security mechanism, in terms of its basic functionality and in terms of its security properties, the latter defined in terms of **game(s)**.
- ▶ Then try to see if it is **realizable** (perhaps with additional assumption, and/or the use of previously defined mechanisms – e.g. can you realize a digital signature secure against existential forgery based on adaptive chosen message attack using a public key encryption scheme that is semantically secure?).
- ▶ It is **OK** to have many different variant definitions for same intuitive notion.



Other examples

Consider the following cryptographic applications. Can you define appropriate games?

- ▶ **Proxy re-encryption:** Alice tells proxy how to transform messages encrypted in PK_{Alice} into ones encrypted with PK_{Bob} (without decrypting first!).

Other examples

Consider the following cryptographic applications. Can you define appropriate games?

- ▶ **Proxy re-encryption:** Alice tells proxy how to transform messages encrypted in PK_{Alice} into ones encrypted with PK_{Bob} (without decrypting first!).
- ▶ **Fully homomorphic encryption:** Alice gives Bob encrypted data, and later instructs him how to compute encrypted function of data; Bob learns nothing about data or result.

Other examples

Consider the following cryptographic applications. Can you define appropriate games?

- ▶ **Proxy re-encryption:** Alice tells proxy how to transform messages encrypted in PK_{Alice} into ones encrypted with PK_{Bob} (without decrypting first!).
- ▶ **Fully homomorphic encryption:** Alice gives Bob encrypted data, and later instructs him how to compute encrypted function of data; Bob learns nothing about data or result.
- ▶ **Commitments:** commit now, reveal (“open”) later.
 - ▶ **Binding:** Once committed, sender can only open one way.
 - ▶ **Hiding:** Recipient can't tell what was committed to.

Other examples

Consider the following cryptographic applications. Can you define appropriate games?

- ▶ **Proxy re-encryption:** Alice tells proxy how to transform messages encrypted in PK_{Alice} into ones encrypted with PK_{Bob} (without decrypting first!).
- ▶ **Fully homomorphic encryption:** Alice gives Bob encrypted data, and later instructs him how to compute encrypted function of data; Bob learns nothing about data or result.
- ▶ **Commitments:** commit now, reveal (“open”) later.
 - ▶ **Binding:** Once committed, sender can only open one way.
 - ▶ **Hiding:** Recipient can't tell what was committed to.
- ▶ **Anonymous Credentials:** Alice can prove to third party she has valid credential from issuer X, without revealing her identity or showing credential to third party.



Reusability and Composability

- ▶ A useful security mechanism should be securely usable in larger contexts:
 - ▶ repeatedly
 - ▶ concurrently with other mechanisms or other copies of itself.
- ▶ Games can also be used to handle such extensions (see Canetti's "Universal Composability" model).



Non-cryptographic security properties

- ▶ Can we similarly provide good definitions of **non-cryptographic** security properties?
- ▶ Let's try some examples...

Example: biometric identification

- ▶ Can adversary defeat biometric identification method?
(That is, pass himself off as someone else?)
- ▶ Typical definitions are already “game-like”, defining “false acceptance rate”, etc...

Example: Program Obfuscation

- ▶ Can one “obfuscate” a program in such a way that it retains its functionality (that is, you can run it), but no other useful information can be gleaned by looking at the code?
- ▶ (Cryptographers have taken a stab at defining this, with limited success...)



Example: Sandboxing

- ▶ Can a program tell if it is **running in a sandbox**?

Example: Sandboxing

- ▶ Can a program tell if it is **running in a sandbox**?
- ▶ Game must specify what operations program can do (system calls? network?)



Example: Sandboxing

- ▶ Can a program tell if it is **running in a sandbox**?
- ▶ Game must specify what operations program can do (system calls? network?)
- ▶ This is an **indistinguishability** game...

Summary

- ▶ A “science of security” must be founded on **good definitions** for security properties.
- ▶ Cryptographers typically define a security property in terms of a **game** that the adversary must play.
- ▶ This approach may be useful for non-cryptographic security properties.



The End

(Some references given on next slide...)

References

-  Shafi Goldwasser and Silvio Micali.
Probabilistic encryption.
JCSS, 28:270–299, 1984.
Conf. version in STOC'82.
-  Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest.
Digital signature scheme secure against adaptive chosen message attacks.
SIAM J. Computing, 17(2):281–308, 1988.
-  Jonathan Katz and Yehuda Lindell.
Introduction to Modern Cryptography.
Chapman, 2008.
(esp. Section 1.4).
-  Victor Shoup.
Sequences of games, 2006.
<http://www.shoup.net/papers/games.pdf>.