Security Reduction

Outline

- Modern Cryptography
- How to Define Algorithms
- How to Define Security Models
- Brief Introduction to Security Reduction

Classical Cryptography vs Modern Cryptography

Introduction to Security Reduction

Classical Cryptography vs Modern Cryptography

- Classical cryptography was referred to as art. After the late of 20 centry, cryptography was studied as science and called modern cryptography.
- Modern cryptography is distinguished from classical cryptography by its emphasis on definitions, models and proofs.
  - Definitions of algorithms, hard assumptions, advantage and so on.
  - Computation model, security model and so on.
  - Security reduction, game-hopping proof and so on.

Introduction to Security Reduction

Clarified Concepts

- Cryptography, such as public-key cryptography, group-based cryptography, and elliptic-curve cryptography, is a security mechanism to provide security services for authentication, confidentiality, integrity, etc.
- Cryptosystem, such as digital signatures, public-key encryption, and identity-based encryption, is a suite of algorithms that provides ONE security service or more.
- Scheme, such as the BLS signature scheme[1], is a specific construction or implementation of the corresponding algorithms for ONE cryptosystem.

[1] Dan Boneh, Ben Lynn, Hovav Shacham (2004). ”Short Signatures from the Weil Pairing”. Journal of Cryptology. 17: 297-319. doi:10.1007/s00145-004-0314-9.

Research Direction

To solve a problem with a new scheme[2], you need to know

- How to define algorithms for a cryptosystem.
- How to define a security model for a cryptosystem.
- How to construct a scheme for a cryptosystem.
- How to prove security for the proposed scheme.

[2] A new scheme for an existing cryptosystem or a new one.

How to Define Algorithms

How to Define Algorithms for a Cryptosystem?

- Fully understand the security service(s) (motivation).
- Which entities (not names) are involved.
- How many algorithms are involved.
- What is the name of each algorithm.
- Who runs each algorithm.
- What are inputs and outputs of each algorithm.

How to Define Algorithms for Digital Signatures?

Motivation

- A party, say Alice, wants to convince all other parties that a message m is published by her.
- To do so, Alice generates a public/secret key pair (pk, sk) and publishes the public key pk to all verifiers.
- To generate a signature σ_m on m, she digitally signs m with her secret key sk.
- Upon receiving (m, σ_m), any receiver who already knows pk can verify the signature σ_m and confirm the origin of the message m.

How to Define Algorithms for Digital Signatures?

Motivation

- A party, say Alice, wants to convince all other parties that a message m is published by her.
- To do so, Alice generates a public/secret key pair (pk, sk) and publishes the public key pk to all verifiers.
- To generate a signature σ_m on m, she digitally signs m with her secret key sk.
- Upon receiving (m, σ_m), any receiver who already knows pk can verify the signature σ_m and confirm the origin of the message m.

Which entities (not names) are involved.

Signer, Verifier

How to Define Algorithms for Digital Signatures?

Motivation

- A party, say Alice, wants to convince all other parties that a message m is published by her.
- To do so, Alice generates a public/secret key pair (pk, sk) and publishes the public key pk to all verifiers.
- To generate a signature σ_m on m, she digitally signs m with her secret key sk.
- Upon receiving (m, σ_m), any receiver who already knows pk can verify the signature σ_m and confirm the origin of the message m.

How many algorithms are involved.

1 (system parameter generation) + 3

How to Define Algorithms for Digital Signatures?

Motivation

- A party, say Alice, wants to convince all other parties that a message m is published by her.
- To do so, Alice generates a public/secret key pair (pk, sk) and publishes the public key pk to all verifiers.
- To generate a signature σ_m on m, she digitally signs m with her secret key sk.
- Upon receiving (m, σ_m), any receiver who already knows pk can verify the signature σ_m and confirm the origin of the message m.

What is the name of each algorithm.

SysGen, KeyGen, Sign, Verify

How to Define Algorithms for Digital Signatures?

Motivation

- A party, say Alice, wants to convince all other parties that a message m is published by her.
- To do so, Alice generates a public/secret key pair (pk, sk) and publishes the public key pk to all verifiers.
- To generate a signature σ_m on m, she digitally signs m with her secret key sk.
- Upon receiving (m, σ_m), any receiver who already knows pk can verify the signature σ_m and confirm the origin of the message m.

Who runs each algorithm.

SysGen (Authority), KeyGen (Signer), Sign (Signer), Verify (Verifier)

A signer can generate a system parameter for him/her to use alone so that SysGen is run by Signer.

How to Define Algorithms for Digital Signatures?

SysGen(λ)$\rightarrow SP$.

KeyGen(SP)$\rightarrow (pk, sk)$.

Sign(SP, sk, m) $\rightarrow \sigma_m$.

Verify(m, σ_m, SP, pk)$\rightarrow \{0, 1\}$.

input and output only for algorithm definition.

How to Define Algorithms for Digital Signatures?

SysGen: The system parameter generation algorithm takes as input a security parameter λ. It returns the system parameters SP.

KeyGen: The key generation algorithm takes as input the system parameters SP. It returns a public/secret key pair (pk, sk).

Sign: The signing algorithm takes as input a message m from its message space, the secret key sk, and the system parameters SP. It returns a signature of m denoted by σ_m.

Verify: The verification algorithm takes as input a pair (m, σ_m), the public key pk, and the system parameters SP. It returns “accept” if σ_m is a valid signature of m signed with sk; otherwise, “reject.”

input and output only for algorithm definition.

Introduction to Security Reduction

How to Define Security Models

Questions Before Security Model

- How to analyze the security of a scheme?
- If a scheme is secure in a security model, can the scheme resist any attack?
- Is a security model defined for a scheme?

What is Security Model?

- When we propose a scheme for a cryptosystem, we do not analyze its security against a list of attacks, such as replay attack and collusion attack. Instead, we analyze that the proposed scheme is secure in a security model.
- A security model can be seen as an abstract of multiple attacks for a cryptosystem. If a proposed scheme is secure in a security model, it is secure against any attack that can be described and captured in this security model.
- Abstracted attacks focus on what information the adversary learns instead of how the adversary attacks.
- A security model is defined for a cryptosystem not for a scheme.

Security Model

- A security model can be seen as an abstract of multiple attacks for a cryptosystem.
- To model the security for a cryptosystem, challenger interacts with an adversary.
- A security model can be seen as a game (interactively) played between the challenger and the adversary.
- Components：

  - ✓ What information the adversary can query.
  - ✓ When the adversary can query information.
  - ✓ How the adversary wins the game (breaks the scheme).

How to Define a Security Model?

We can use a game played between adversary and challenger to describe a security model.

- The challenger is the secret key owner of a cryptosystem.
- The adversary is trying to break the cryptosystem.

A security model defines:

- The adversary's capabilities:
  - What information the adversary can query
  - When the adversary can query information
- The adversary's security goal:
  - How the adversary wins the game (breaks the scheme).

How to Define a Security Model?

The definition of a security model is composed of four parts:

- Setup: The initialization between the adversary and the challenger.
- Capabilities: Describe what and when the adversary queries.
- Security Goal: The condition of wining the game for the adversary.
- Advantage: Define a parameter satisfying.
  - If the parameter is negligible, the cryptosystem is secure.
  - Otherwise, the parameter is non-negligible and it is insecure.

Introduction to Security Reduction

When Defining a Security Model

When defining a security model, we consider input and output only the same as algorithm definition. The definition must be applicable to all schemes (proposed for a cryptosystem).

For example, when the adversary queries a signature on a message m, the challenger must respond to this query by running the signing algorithm with (pk, sk, m) as input to output signature σ_m.

- We don't care how to generate the signature σ_m.
- We don't care what the signature looks like [3].
- We only care what the adversary queried (i.e. the message m)

[3] The detailed binary representation. The signature is just a symbol denoted by σ_m.

When Defining a Security Model

When defining a security model, we don’t consider

- The adversary’s trivial attack that can help the adversary break the cryptosystem easily. For example, the adversary asks the challenger to share all secrets (e.g. all secret keys) with it.
- The adversary’s strategy about how it obtains a piece of information. Therefore, when the adversary makes a query, the challenger must respond to this query correctly and honestly.

When defining a security model, make sure that

the advantage is negligible.

(Therefore, the adversary is restricted in some queries.)

Security Model for Digital Signatures

- Setup: A key pair is generated and the adversary knows pk.
- Capabilities: The adversary obtains signatures of some messages.
- Security Goal: The adversary cannot forge a new signature.
- Advantage: The probability of successful forgery.

Trivial Attack: The adversary queries the signing/secret key.

The advantage is 1.

Security Model of Digital Signatures

The security model of existential unforgeability against chosen-message attacks (EU-CMA) can be described as follows.

Setup. Let SP be the system parameters. The challenger runs the key generation algorithm to generate a key pair (pk, sk) and sends pk to the adversary. The challenger keeps sk to respond to signature queries.

Query. The adversary makes signature queries on messages that are adaptively chosen by the adversary itself. For a signature query on the message m_i, the challenger runs the signing algorithm to compute σ_{m_i} and then sends it to the adversary.

Forgery. The adversary returns a forged signature σ_{m^*} on some m^* and wins the game if

- σ_{m^*} is a valid signature of the message m^*.
- A signature of m^* has not been queried in the query phase.

The advantage ϵ of winning the game is the probability of returning a valid forged signature.

Notes on Security Model

- A cryptosystem might have more than one security service.
- Each security service needs one security model.
- For example:

signcryption = signature + encryption

Two security models:
cannot break signature,   cannot break encryption

Notes on Security Model

- Security models for a security service of a cryptosystem might have different definitions, depending on capabilities and security goal.
  - Standard security model: the capability and security goal are acceptable by most applications.
  - Strong security model: the adversary has a strong capability and/or a easier security goal than the standard one. (Resist more attacks)
  - Weak security model: the adversary has a weak capability and/or a harder security goal than the standard one. (Resist less attacks)

- For example: the adversary is only allowed to know signatures on some given messages, instead of querying signatures on any messages.

Notes on Security Model

- A scheme secure in a strong security model means that it can resist more powerful attacks that a scheme in a weak security model.
- Strong security model was proposed due to some applications that need strong security.
- Weak security model was proposed because
  - a weak security is enough and we can construct more efficient schemes in this model.
  - some very efficient schemes can only be proved security in the weak security model.

Introduction to Security Reduction

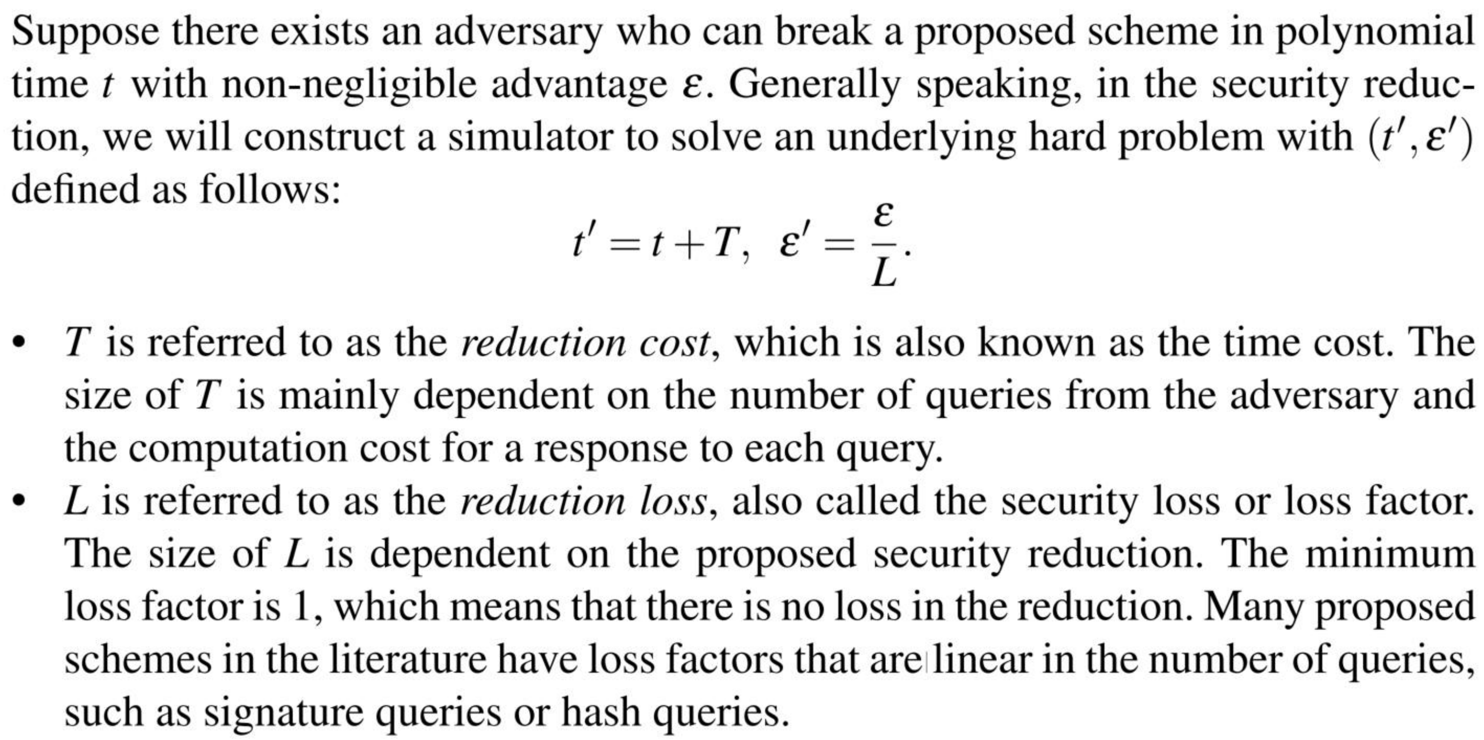
Security Reduction

- Concept: the process from insecure to easy in the proof by contradiction
- A security reduction works if we can find a solution to a problem instance of the mathematical hard problem with the help of the adversary's attack.
- The core and difficulty of the security reduction is to generate such a different but well-prepared scheme.

✓ real scheme, challenger, and real attack

✓ simulated scheme, simulator, and simulation

Reduction Cost and Reduction Loss



Suppose there exists an adversary who can break a proposed scheme in polynomial time t with non-negligible advantage ε. Generally speaking, in the security reduction, we will construct a simulator to solve an underlying hard problem with (t', ε') defined as follows:

$$t' = t + T, \;\; \varepsilon' = \frac{\varepsilon}{L}.$$

- T is referred to as the *reduction cost*, which is also known as the time cost. The size of T is mainly dependent on the number of queries from the adversary and the computation cost for a response to each query.
- L is referred to as the *reduction loss*, also called the security loss or loss factor. The size of L is dependent on the proposed security reduction. The minimum loss factor is 1, which means that there is no loss in the reduction. Many proposed schemes in the literature have loss factors that are linear in the number of queries, such as signature queries or hash queries.

✓ Loose: L is at least linear in the number of queries

vs

✓ Tight: L is a constant number or is small (e.g.,sub-linear in the number of queries).

Ideal Security Reduction

- Security Model.
- Hard Problem.
- Reduction Cost and Reduction Loss.
- Computational Restrictions on Adversary.

Probability and Advantage

- Probability: the measure of the likelihood about a successful attack on a scheme or a correct solution to a problem instance.
- Advantage is the measure of how successfully an attack algorithm can break a proposed scheme or a solution algorithm can solve a problem, compared to the idealized probability P_{ideal} in the corresponding security model.

  - If P_{ideal} is non-negligible, we define the advantage as

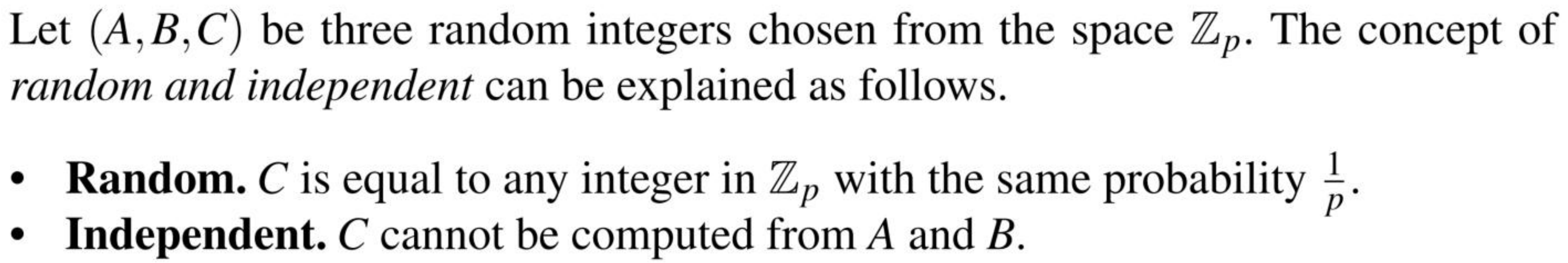
$$\text{Advantage} = \text{Probability of Successful Attack} \; - \; P_{ideal}.$$

  - If P_{ideal} is negligible, we define the advantage as

$$\text{Advantage} = \text{Probability of Successful Attack}.$$

Random and Independent

- The simulation is indistinguishable from the real attack
- In a simulated scheme, if random numbers are not randomly chosen, but generated from a function, we must prove that these simulated random numbers generated by the function are also random and independent from the point of view of the adversary.



Let (A, B, C) be three random integers chosen from the space $\mathbb{Z}_p$. The concept of *random and independent* can be explained as follows.

- **Random.** C is equal to any integer in $\mathbb{Z}_p$ with the same probability $\frac{1}{p}$.
- **Independent.** C cannot be computed from A and B.