DIGITAL SIGNATURES AND AUTHENTICATION

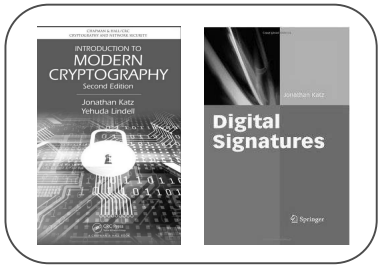
数字签名与认证

sd04630200

课程信息

课序号：sd04630200
任课教师：袁泉，yuanquan@sdu.edu.cn
时间地点：周一12，会文南楼225
课时：16*2　　学分：2

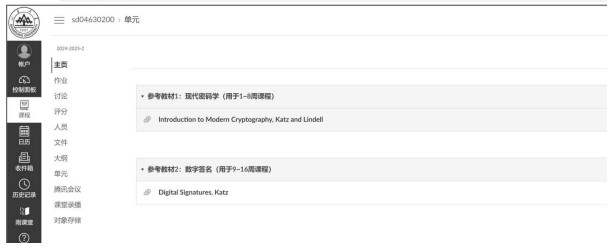
平时成绩50%：课堂作业 + 课后作业 + 期中
期末成绩50%



教材及参考文献

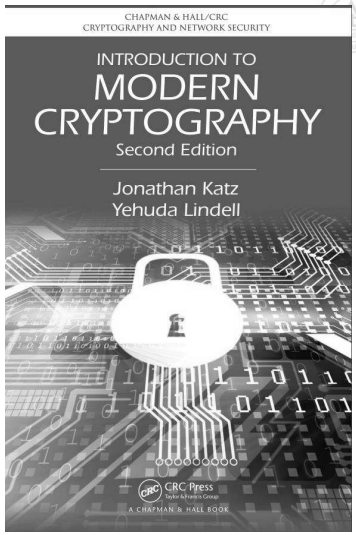
A. Introduction to Modern Cryptography
   - Jonathan Katz, Yehuda Lindell

B. Digital Signatures
   - Jonathan Katz



课程大纲 （上半部分 教材A 1~8周）

- 第一章 现代密码学简介 (A. Sec 1 Sec 3.1, 3.2)
- 第二章 消息认证码与抗碰撞Hash函数 (A. Sec 4~5)
  - 第一节 消息认证码定义及安全模型 (4.1, 4.2)
  - 第二节 可变长度消息认证码 (4.3)
  - 第三节 Hash函数 (5.1, 5.4, 5.6)
  - 第四节 Merkle-Damgard转换及HMAC (5.2, 5.3)
- 第三章 单向函数 (A. Sec. 7)
  - 第一节 单向函数与硬核谓词 (7.1, 7.3)
  - 第二节 伪随机数生成器 (7.4)
  - 第三节 伪随机函数 (7.5)
- 第四章 公钥数字签名 （下半部分 教材B 9~16周）



- QQ群号：819343089
  验证问题：数字签名与认证
  通知，作业，课件（实时更新）

- 关于教室
- 关于平时成绩与考试
- 关于内容

课时一 现代密码学的三大法则

- Three main principles that distinguish modern cryptography from the classical cryptography
  - **Principle 1 Formulation of Exact Definitions**
  - **Principle 2 Reliance on Precise Assumptions**
  - Principle 3 Rigorous Proofs of Security

什么是密码学？

- Password？Passwordology？(×)

什么是密码学？

- ~~Password？Passwordology？~~ (×)
- Encrypt & Cryptology!

  研究（加密等）密码算法的科学

Cryptology 密码学
- Cryptanalysis 密码分析
- Cryptography 密码构造

什么是密码学？

- ~~Password？Passwordology？~~ (×)
- Encrypt & Cryptology!

  研究（加密等）密码算法的科学

Cryptology 密码学
- Cryptanalysis 密码分析
- Cryptography 密码构造

（粗略翻译为密码学）

Modern Cryptography

- What is CRYPTOGRAPHY
  - The scientific study of techniques for securing digital information, transactions, and distributed computations

《中华人民共和国密码法》2020年1月1日

《中华人民共和国电子签名法》2005年4月1日

“当事人约定使用电子签名、数据电文的文书，不得仅因为其采用电子签名、数据电文的形式而否定其法律效力。”

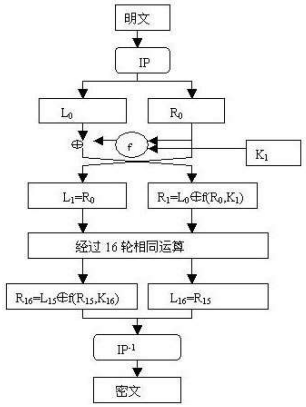
如何设计安全的密码算法？

- 思路一：

  先设计，再分析

  DES,…

  当密码算法能抵抗现有的攻击方法时，我们认为该算法是安全的



Cryptanalysis

密码分析

Differential Attacks

差分攻击

Linear Attacks

线性攻击

如何设计安全的密码算法？

- 思路一：

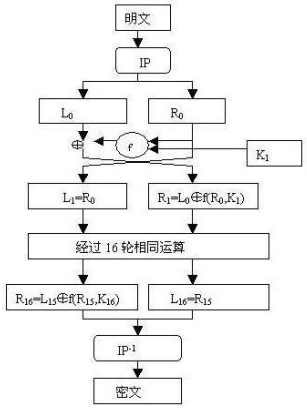
  先设计，再分析

  DES,…

  当密码算法能抵抗**现有**的攻击方法时，我们认为该算法是安全的

  隐患：未来的攻击！

  更**可靠**的安全性？



Cryptanalysis

密码分析

Differential Attacks

差分攻击

Linear Attacks

线性攻击

如何设计安全的密码算法？

- 思路二：

  可证明安全性

  在构造密码算法的同时，运用数学方法给出其**安全性证明**

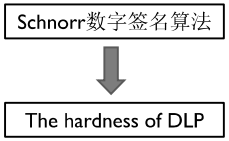
现代密码学 Modern Cryptography

可证明安全

- 主要思路：归约证明（Reduction）（反证法）

  例：Schnorr数字签名算法

  其安全性可以归约于求解离散对数（DLP）的困难性：



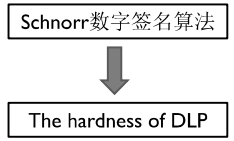
可证明安全

- 主要思路：归约证明（Reduction）（反证法）

  例：Schnorr数字签名算法

  其安全性可以归约于求解离散对数（DLP）的困难性：

  如果敌手能破坏Schnorr数字签名算法的安全性，
  则该敌手就可以**用来**求解离散对数问题



可证明安全

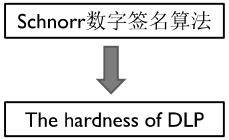
- 主要思路：归约证明（Reduction）（反证法）

  例：Schnorr数字签名算法

  其安全性可以归约于求解离散对数（DLP）的困难性：

  如果敌手能破坏Schnorr数字签名算法的安全性，
  则该敌手就可以**用来**求解离散对数问题

  又因为求解离散对数问题**是困难的**
  所以**不存在**上述敌手
  Schnorr数字签名算法是**可证明安全的**

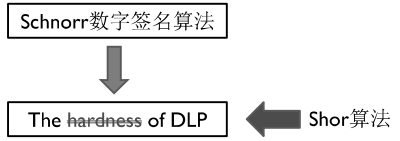


可证明安全

- 有可证明安全的密码算法是不是绝对的安全？

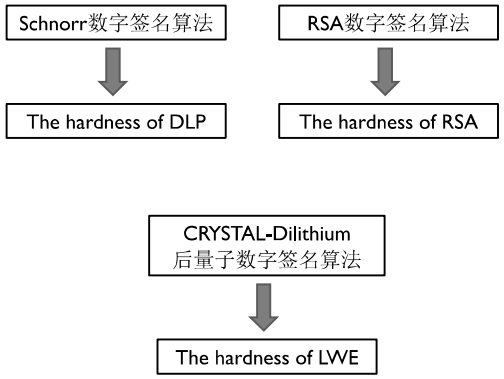
  不一定！

  例：Schnorr数字签名算法
  其安全性可以归约于求解离散对数（DLP）的困难性
  但量子算法Shor**可以**求解DLP问题
  所以，当量子计算机被完全实现时
  “DLP的困难性”这一基础假设**不成立**



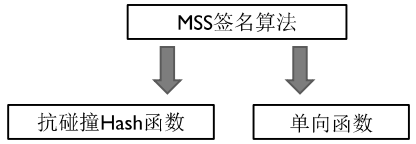
可证明安全

- 不同的数字签名可归约到不同的安全性假设
- 安全性假设多种多样：
  - 经典的数学困难问题
    - DLP问题，RSA问题（经典密码学）
    - SIS问题，LWE问题（后量子密码学）



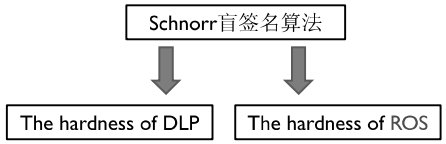
可证明安全

- 不同的数字签名可归约到不同的安全性假设
- 安全性假设多种多样：
  - 经典的数学困难问题
    - DLP问题，RSA问题（经典密码学）
    - SIS问题，LWE问题（后量子密码学）
  - 其他**基础**密码原语
    - 抗碰撞Hash函数，单向函数，伪随机函数……



可证明安全

- 不同的数字签名可归约到不同的安全性假设
- 安全性假设多种多样：
  - 经典的数学困难问题
    - DLP问题，RSA问题（经典密码学）
    - SIS问题，LWE问题（后量子密码学）
  - 其他**基础**密码原语
    - 抗碰撞Hash函数，单向函数，伪随机函数……
  - 非典型性的困难问题 & 多种安全性假设的混合
    - 具体实例具体分析



现代密码学三大法则

- Three main principles that distinguish modern cryptography from the classical cryptography
  - Principle 1 Formulation of Exact Definitions
  - **Principle 2 Reliance on Precise Assumptions**
  - Principle 3 Rigorous Proofs of Security

现代密码学三大法则：安全性假设

- **安全性假设必须要被明确定义**

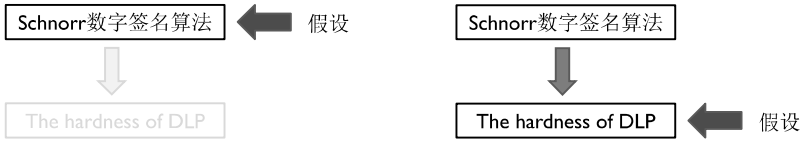
Reliance on Precise Assumptions

- 假设的真实性 Validation of the assumption
  - 假设是无法被证明的但被**认为**是正确的。—— 类比数学证明中的公理
    Assumptions are statements that are not proven but are rather conjectured to be true.
  - 假设被研究得越多（且没有被成功反驳），则该假设的可信度越高
    The more the assumption is looked at without being successfully refuted, the more confident we are that the assumption is true.

    如：对DLP问题的研究已有50年历史，但依然（在经典计算机下）无法求解。

可证明安全

- 为什么要将算法安全性归约到安全性假设，而不是直接假设算法本身是安全的？



可证明安全

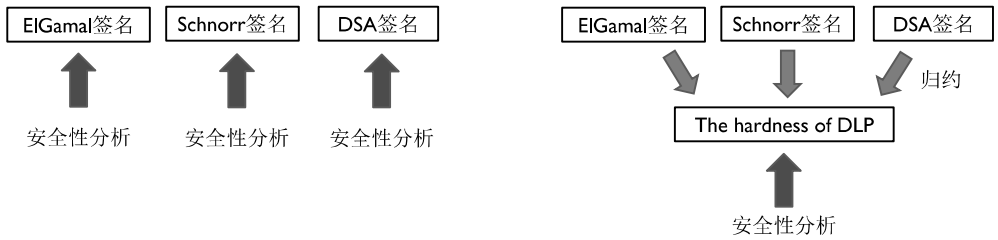
- 为什么要将算法安全性归约到安全性假设，而不是直接假设算法本身是安全的？

1. 安全性假设更容易被陈述和研究
2. 安全性假设一般经历了更长久的时间检测，可信度更高

可证明安全

- 为什么要将算法安全性归约到安全性假设，而不是直接假设算法本身是安全的？

1. 安全性假设更容易被陈述和研究
2. 安全性假设一般经历了更长久的时间检测，可信度更高
3. 同一安全性假设可以用于**多个算法**中



现代密码学三大法则

- Three main principles that distinguish modern cryptography from the classical cryptography
  - **Principle 1 Formulation of Exact Definitions**
  - Principle 2 Reliance on Precise Assumptions
  - Principle 3 Rigorous Proofs of Security

现代密码学三大法则：安全定义

- **确切定义的范式化**

  Formulation of Exact Definitions

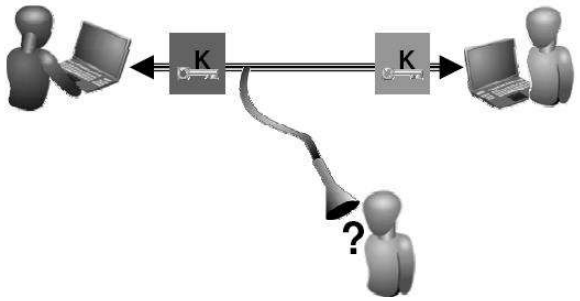
- 如何定义“安全”？

  对敌手（攻击者）的行为进行**清晰**的定义：

  1. Kerckhoffs原则
  2. 攻击目标
  3. 能获取的信息
  4. 计算能力

私钥加密算法

-



$$\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m.$$

如何定义私钥加密算法的安全性？

对敌手（攻击者）的行为进行清晰的定义：

1. Kerckhoffs原则

Kerckhoffs' principle

- Keys and Kerckhoffs' principle
  - 19世纪末 , Auguste Kerckhoffs
    - “The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.”

- 安全性必须仅依赖于密钥的隐私性，而不能依赖于算法的隐私性

$\mathsf{Enc}_k(m)$

暴露 隐藏

WHY?

Kerckhoffs' principle 1: 对密钥的保密优于对算法的保密

■ 可操作性：

保密一个短密钥比保密整个算法更容易。

It is much easier for the parties to maintain secrecy of a short key

■ 可替换性：

一旦密钥被暴露，更换新密钥比更换新算法要更容易。

In case the key is exposed, it is much easier for the honest parties to change the key than to replace the algorithm being used.

■ 可共享性：

多用户使用同一套算法，且使用不同密钥，更容易实现。

It will be significantly easier for all parties to use the same algorithm, but different keys, than for everyone to use a different program

Kerckhoffs' principle 2: 公开算法设计 Open cryptographic Design

■ 原则上需要把所有密码算法的代码及运行原理公开，而非保密。

■ 接受公众对安全性的检测：

Security flaws could be revealed by "ethical hackers" and made public.

Public design enables the establishment of standards.

明确攻击目标：安全模型的定义

对敌手（攻击者）的行为进行清晰的定义：

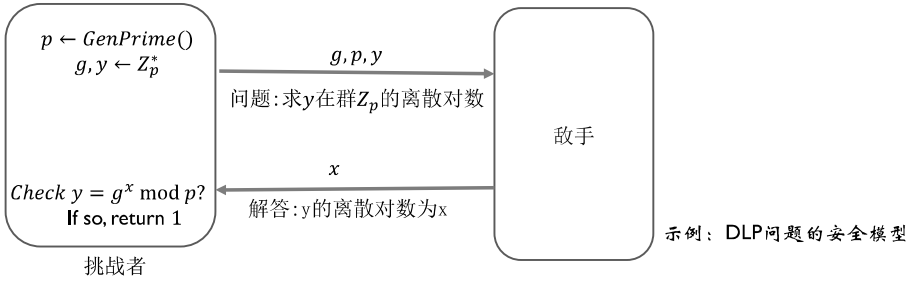
1. Kerckhoffs原则
2. 攻击目标

■ 由**挑战者**和**敌手**双方进行的**交互性**实验/游戏（Experiment/Game）

➢ 挑战者（Challenger）：提出问题，检验问题 （有时会将挑战者弱化为实验本身）

➢ 敌手（Adversary）：解决问题

刻画敌手攻击算法安全性（或破解安全性假设）的**形式**（**不包括解决问题的具体过程**）



明确攻击目标：安全模型的定义

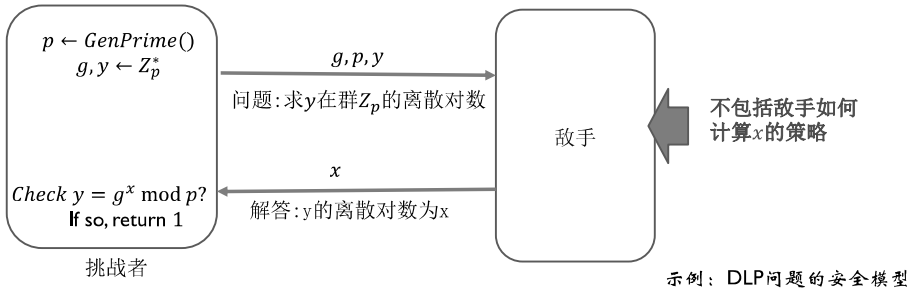
$p \leftarrow GenPrime()$

■ 如何定义安全性（困难性）？

如果**任意敌手**都无法完成安全模型中的任务，

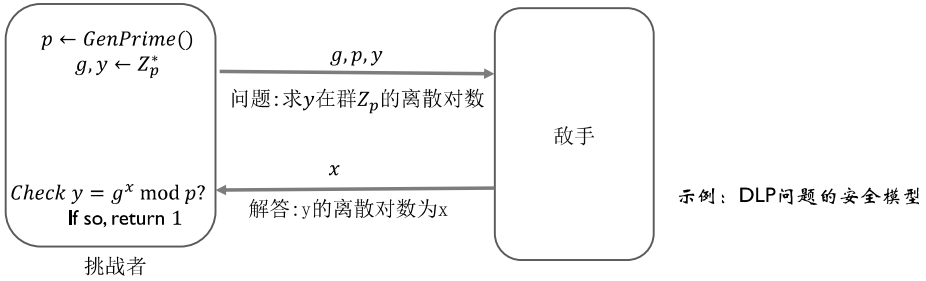
则称该算法（困难问题）是安全的（或困难的）。

■ 例：如果**任意敌手**都无法破解DLP问题，则称DLP问题是困难的。



安全模型的定义

- **在DLP问题的安全模型中，敌手成功与否是由哪些因素决定的？**

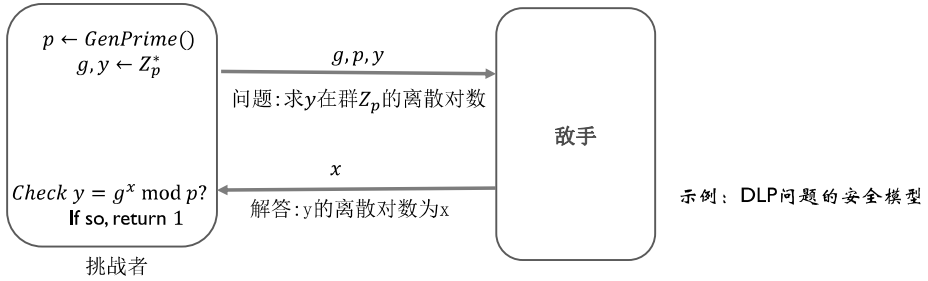


示例：DLP问题的安全模型

安全模型的定义

- **在DLP问题的安全模型中，敌手成功与否是由哪些因素决定的？**

- **敌手的策略**

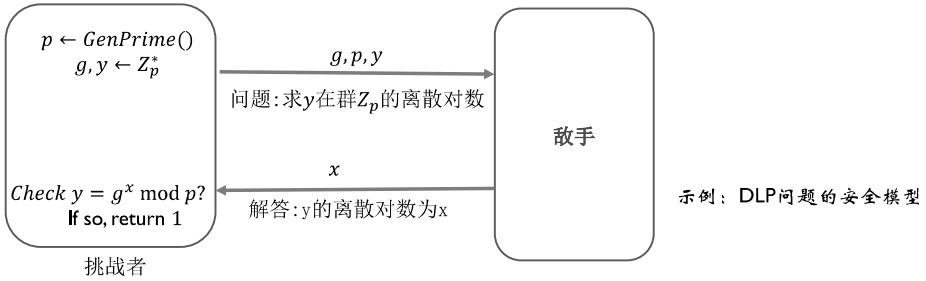


示例：DLP问题的安全模型

安全模型的定义

- **在DLP问题的安全模型中，敌手成功与否是由哪些因素决定的？**
- **敌手的策略**

- **当敌手是随机算法时，敌手随机数的选取**

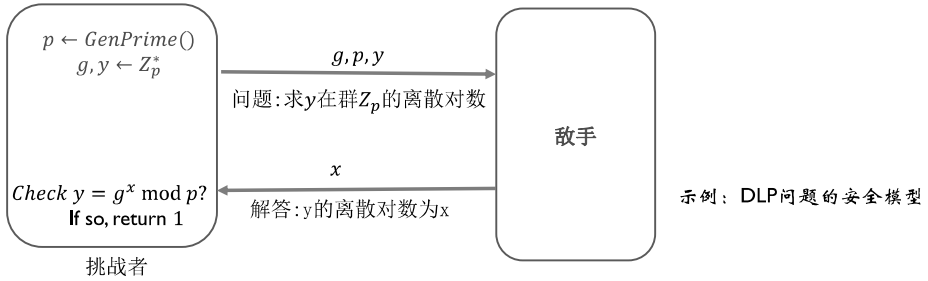


示例：DLP问题的安全模型

安全模型的定义

- **在DLP问题的安全模型中，敌手成功与否是由哪些因素决定的？**
- **敌手的策略**
- **当敌手是随机算法时，敌手随机数的选取**

- **挑战者随机数的选取 （易忽略）**



示例：DLP问题的安全模型

如何定义私钥加密算法的安全性？

■ 如果任意敌手**都做不到**某项任务，则说该算法是安全的。

■ 如果**存在**某敌手**做到了**某项任务，则说该算法是**不**安全的。

- 获取了密钥
- 获取了明文

如何定义私钥加密算法的安全性？

■ 如果任意敌手**都做不到**某项任务，则说该算法是安全的。

- 做不到获取密钥？
- 做不到获取明文？

敌手的目标——反例！

■ 如果敌手无法获取**密钥**，则说该私钥加密算法是安全的？

■ 设计以下128-bit算法：

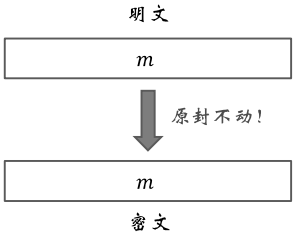
秘钥：$k \leftarrow \{0,1\}^{128}$.

$Enc_k(m) := m$. $Dec_k(c) := c$.

■ 显然满足：对任意明文m, $m = Dec_k(Enc_k(m))$,

且任意敌手都无法得到“密钥”k

■ 但显然 $(KeyGen, Enc, Dec)$ **不是**一个安全的加密算法！



敌手的目标——反例！

■ 如果敌手无法获取**明文**，则说该私钥加密算法是安全的？

■ 假设128比特AES加密算法是安全的，设计以下256-bit算法：

秘钥：$k \leftarrow \{0,1\}^{128}$.

$Enc_k(m)$: Parse $m := m_1||m_2$. Compute $c_2 := AES.Enc_k(m_2)$

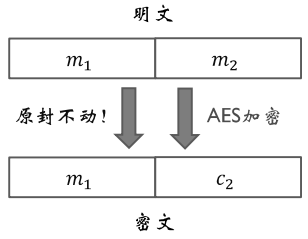
Output $m_1||c_2$.

$Dec_k(c)$: Parse $c = c_1||c_2$. Compute $m_2 := AES.Dec_k(c_2)$

Output $c_1||m_2$.

■ 显然满足：对任意明文m, $m = Dec_k(Enc_k(m))$,

且因为AES是安全的，任意敌手都无法**完整**解密。



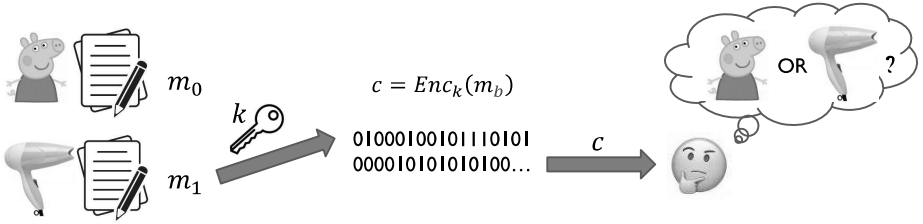
■ 但显然 $(KeyGen, Enc, Dec)$ **不是**一个安全的加密算法！前半部分明文被完全泄露。

如何定义私钥加密算法的安全性？

■ 如果任意敌手**都做不到**某项任务，则说该算法是安全的。

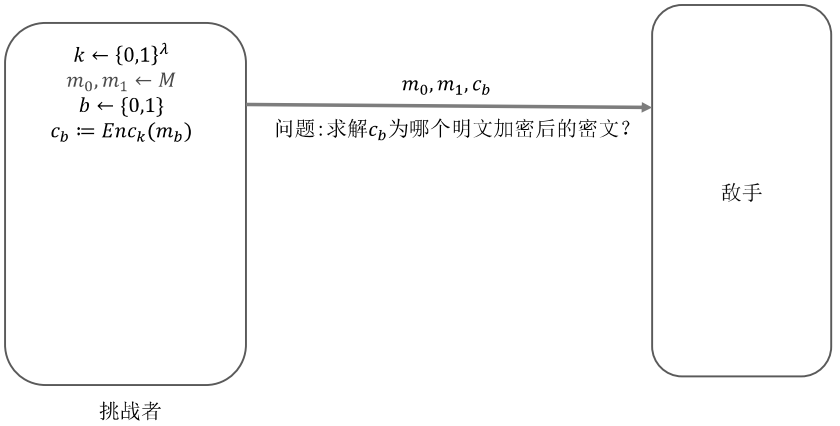
- 做不到获取密钥？
- 做不到获取明文？

- **正确定义：不可区分性 Indistinguishability Experiment**
- 思路：敌手无法区分m_0, m_1的密文



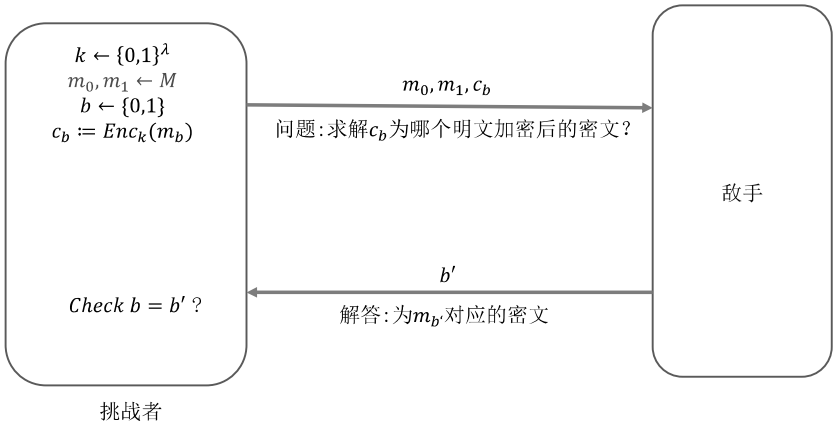
安全模型的定义

■ 不可区分性的安全模型一：**随机挑战下的不可区分性**



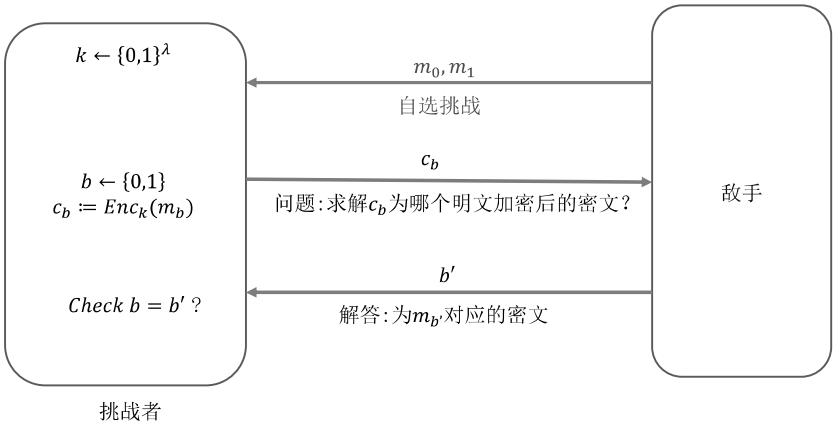
安全模型的定义

■ 不可区分性的安全模型一：**随机挑战下的不可区分性**



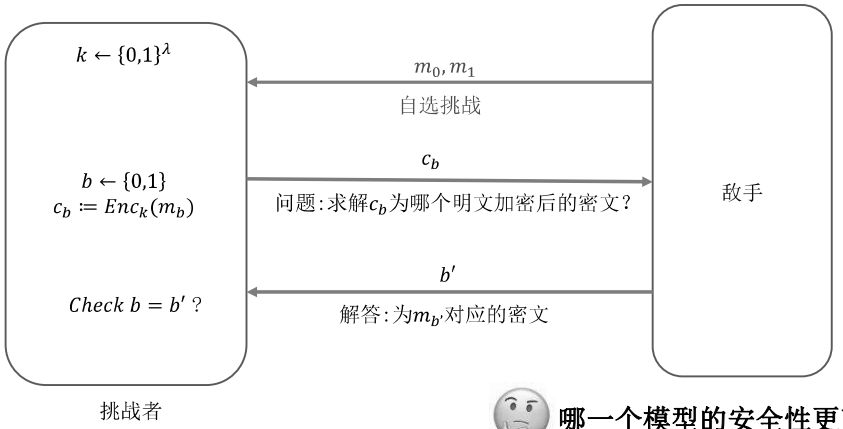
安全模型的定义

■ 定义方法二：**敌手自选挑战下的不可区分性**



安全模型的定义

■ 定义方法二：**敌手自选挑战下的不可区分性**



哪一个模型的安全性更高？

敌手获取的信息

对敌手（攻击者）的行为进行清晰的定义：

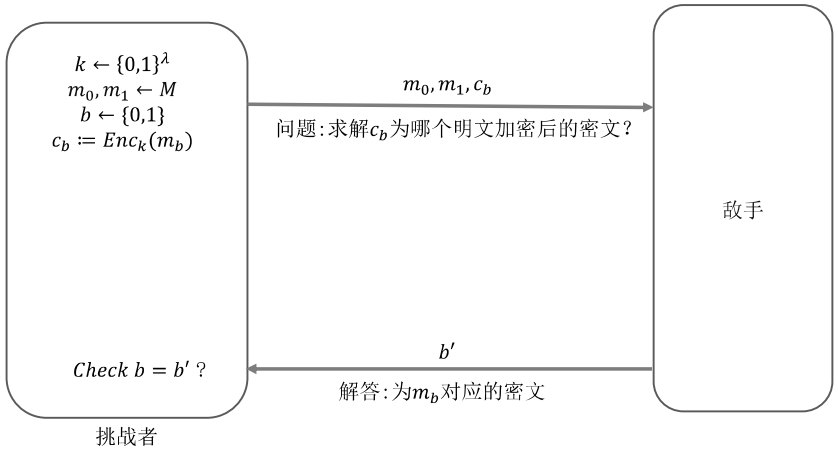
1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息

■ 敌手能获取什么信息？

- 唯密文攻击
  敌手能获取某个（些）密文
- 已知密文攻击
  敌手额外知晓某些明密文对
- 选择明文攻击
  敌手额外拥有加密机，可以加密任意明文（目标明文除外）
- 选择密文攻击
  敌手额外拥有解密机，可以解密任意密文（目标密文除外）

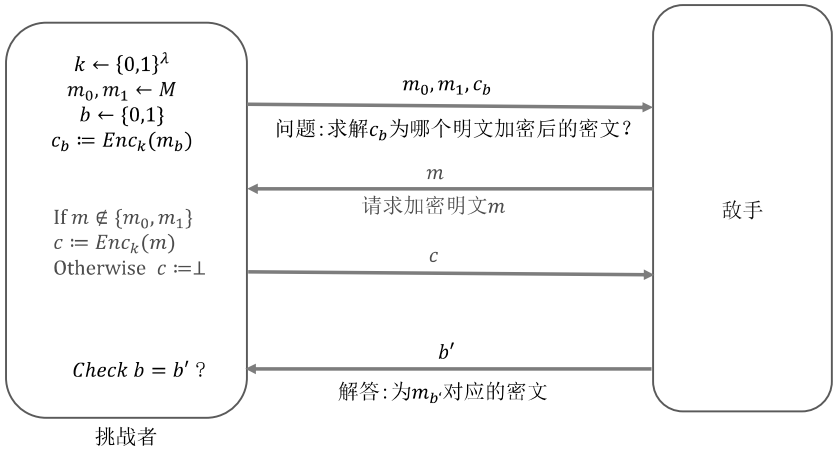
敌手获取的信息

■ 例：**随机挑战下的不可区分性**



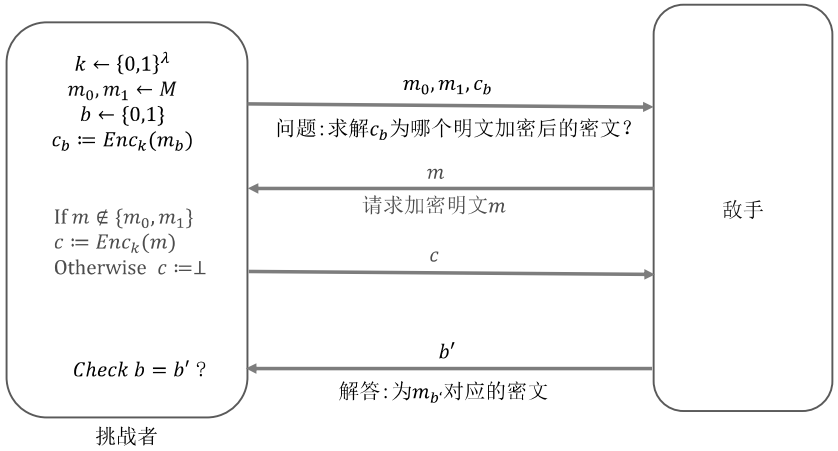
敌手获取的信息

■ 例：**选择明文攻击下随机挑战下的不可区分性**



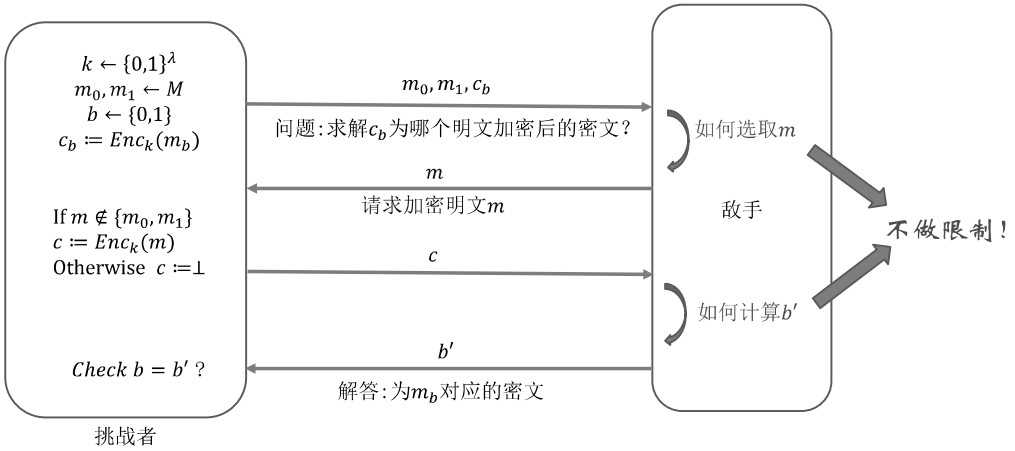
敌手获取的信息

■ 例：选择明文攻击下随机挑战下的不可区分性



敌手获取的信息

■ 例：选择明文攻击下随机挑战下的不可区分性



敌手的能力

对敌手（攻击者）的行为进行清晰的定义：

1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息
4. 计算能力

■ 为什么要强调敌手的能力？

已知明文攻击下的暴力破解：

给定明密文对m和c，穷举k，判断是否满足 $Enc_k(m) = c$. 如是则得到密钥k。

期望上需要$O(2^\lambda)$次Enc计算

暴力破解一般是无法防御的，但也是无法实现的

计算安全性：有效时间的敌手

敌手的能力：计算安全性 Computational Security

■ 只需在**有效时间内**运行的敌手下保证安全性

■ Security is only guaranteed against **efficient** adversaries that run for some feasible amount of time.

■ 敌手可能可以**以极低的概率**成功

■ Adversaries can potentially succeed (i.e., security can potentially fail) with some **very small probability**.

■ 定义方式：确定性或渐进性（Concrete/Asymptotic Approaches）

敌手的能力：计算安全性 Computational Security

确定性定义 Concrete Approach

A scheme is (t, ε)-secure *if any adversary running for time at most* t *succeeds in breaking the scheme with probability at most* ε.

- 例：$(2^{80}, 2^{-30})$-security
- The fastest supercomputer would take about 2 years to operate 2^{80} cycles floating point operations.
- An event that occurs once every hundred years can be roughly estimated to occur with probability 2^{-30} in any given second.

- 缺点：Difficult to provide!

敌手的能力：计算安全性 Computational Security

确定性定义 Concrete Approach

渐进性定义 Asymptotic Approach

- 基于**安全参数**进行定义，用**复杂度（Complexity）**而非确切数值刻画安全性。
- 有效的时间：**多项式**时间（Polynomial-time adversary）
- 极低的概率：**可忽略**概率（Negligible probability）

渐进性定义——安全参数 Security Parameters

- 刻画密码算法、敌手的规模，**作为一切算法的输入**（但常被省略）。
- 例：
  - 私钥加密算法——秘钥比特长度
  - RSA公钥数字签名算法——素数模的比特长度

- 作为算法输入时一般使用一**进制**表示法：安全参数为n时，输入为1^n.

渐进性定义——有效算法 Efficient Algorithms

- 当算法可以在**多项式时间**内完成，则称该算法为有效算法
- An algorithm A runs in **polynomial time** if there exists a **polynomial** $p(\cdot)$ such that, for every input $x \in \{0,1\}^*$, the computation of $A(x)$ terminates within **at most** $p(|x|)$ steps. (Here, $|x|$ denotes the length of the string x.)

- 多项式级：$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$.
- 超多项式级（super-polynomial）：指数级（如2^x），准多项式级（如$x^{\log x}$）
- $|1^n| = n$.

- 额外地，一般默认算法为**概率性**算法（Probabilistic polynomial-time，PPT），即允许算法中使用**随机数**（randomness），其中随机数的长度为多项式级

渐进性定义——可忽略概率 Negligible Probability

■ 可忽略函数：对任意多项式p，存在N使得f(n)<1/p(n) 对任意n>N都成立

DEFINITION 3.4 *A function f from the natural numbers to the non-negative real numbers is* negligible *if for every positive polynomial p there is an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.*

单选题 5分 设置

以下哪个函数是可忽略函数?

可忽略函数：对任意多项式p，存在N使得f(n)<1/p(n) 对任意n>N都成立

A $f_1(n) = \frac{1}{n^{10}}$

B $f_2(n) = \frac{1}{2^{10} \cdot n}$

C $f_3(n) = \log(\log n)$

D $f_4(n) = 2^{-n}$

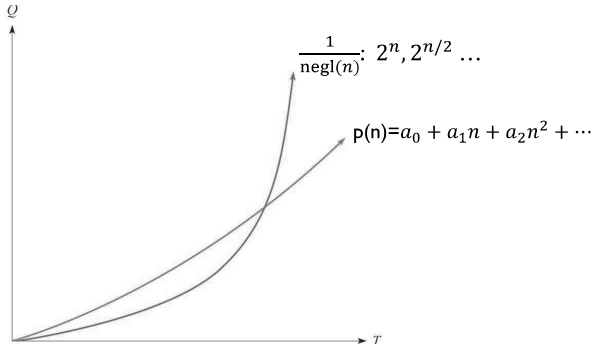
提交

渐进性定义——可忽略概率 Negligible Probability

■ 可忽略函数：对任意多项式p，存在N使得f(n)<1/p(n) 对任意n>N都成立

DEFINITION 3.4 *A function f from the natural numbers to the non-negative real numbers is* negligible *if for every positive polynomial p there is an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.*

■ 例：2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$



渐进性定义——可忽略概率 Negligible Probability

■ 可忽略函数：对任意多项式p，存在N使得f(n)<1/p(n) 对任意n>N都成立

DEFINITION 3.4 *A function f from the natural numbers to the non-negative real numbers is* negligible *if for every positive polynomial p there is an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.*

■ 例：2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$

■ For all constant c, there is an N such that for all integers $n > N$ it holds that $f(n) < n^{-c}$.

渐进性定义——可忽略概率 Negligible Probability

PROPOSITION 3.6 *Let* negl_1 *and* negl_2 *be negligible functions. Then,*

1. *The function* negl_3 *defined by* $\mathsf{negl}_3(n) = \mathsf{negl}_1(n) + \mathsf{negl}_2(n)$ *is negligible.*

■ 例：$2^{-n}+2^{-\sqrt{n}}$

2. *For any positive polynomial* p*, the function* negl_4 *defined by* $\mathsf{negl}_4(n) = p(n) \cdot \mathsf{negl}_1(n)$ *is negligible.*

■ 例： $3n^3 \cdot 2^{-n}$

更常用！

敌手的能力：计算安全性 Computational Security

确定性定义 Concrete Approach

渐进性定义 Asymptotic Approach

对敌手（攻击者）的行为进行
1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息
4. 计算能力

■ 基于**安全参数**进行定义，用**复杂度（Complexity）**而非确切数值刻画安全性。

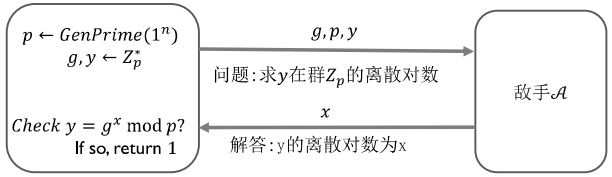
■ 有效的时间：**多项式**时间（Polynomial-time adversary）

■ 极低的概率：**可忽略**概率（Negligible probability）

A scheme is *secure* if for every *probabilistic polynomial-time* adversary $\mathcal{A}$ carrying out an attack of some formally specified type, the probability that $\mathcal{A}$ succeeds in the attack (where success is also formally specified) is *negligible.*

举例：离散对数问题困难性的定义 (更正式定义见Sec 8.3.2)

■ First, we define **discrete logarithm experiment** $\boldsymbol{Dlog_{\mathcal{A}}(n)}$ as follows:



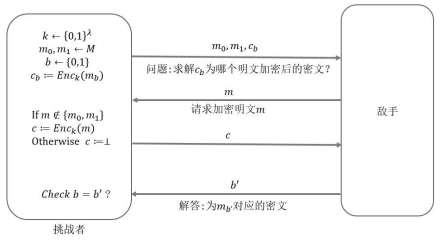
■ We say that **discrete-logarithm problem is hard** if **for every P.P.T adversary**, there exists a **negligible function** $negl(\cdot)$, such that

$$\mathbf{Pr}\left[\boldsymbol{Dlog_{\mathcal{A}}(n) = 1}\right] \leq \boldsymbol{negl(n)},$$

where the probability is taken over the choice of p, g, y and the randomness of $\mathcal{A}$.

选择明文攻击下随机挑战的不可区分性的定义

■ For $\Pi = (Gen, Enc, Dec)$, we define **experiment** $\boldsymbol{RandPrivK^{CPA}_{\Pi,\mathcal{A}}(\lambda)}$ as follows:



■ We say that **Π is secure (or indistinguishable) under chosen message attacks with random challenges** if **for every P.P.T adversary**, there exists a **negligible function** $negl(\cdot)$, such that

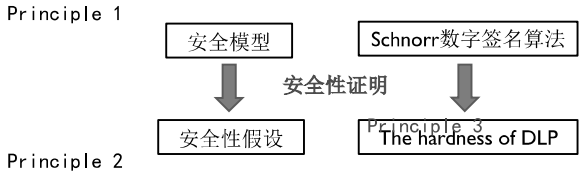
$$\left|\mathbf{Pr}\left[\boldsymbol{RandPriv^{Dlog}_{\Pi,\mathcal{A}}(\lambda) = 1}\right] - \frac{\mathbf{1}}{\mathbf{2}}\right| \leq \boldsymbol{negl(\lambda)},$$

where the probability is taken over the random choices of k, m_0, m_1, b and the randomness of $\mathcal{A}$.

现代密码学三大法则：安全性证明

■ **安全性必须被严格证明**

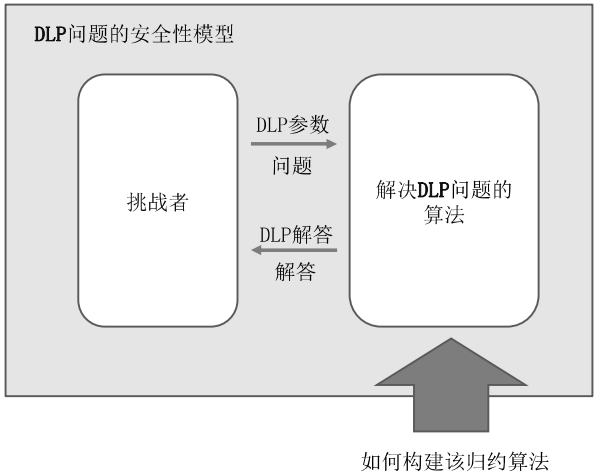
Rigorous Proofs of Security



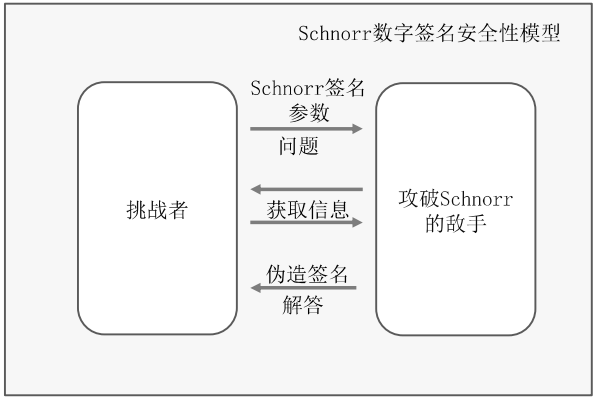
思路：假设敌手能攻击数字签名算法，则可以利用该敌手破解DLP问题

核心：怎么利用？怎么破解？

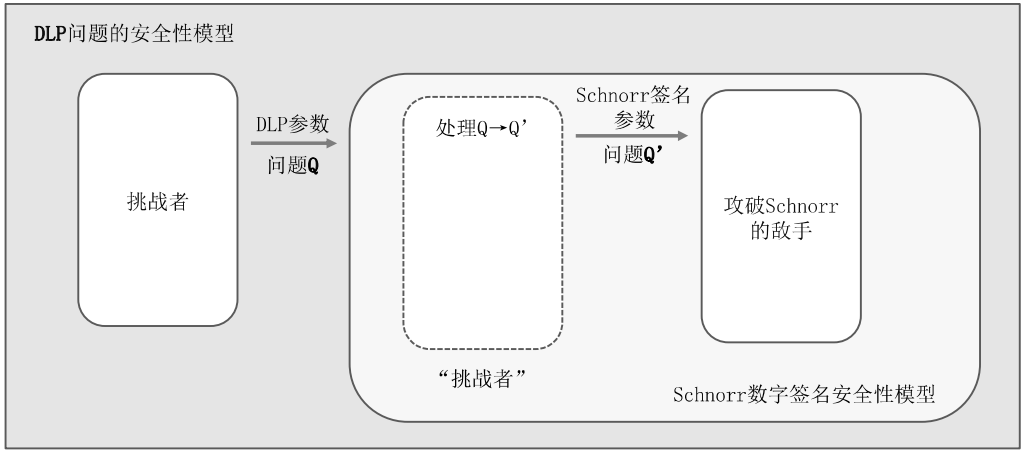
现代密码学三大法则：安全性证明



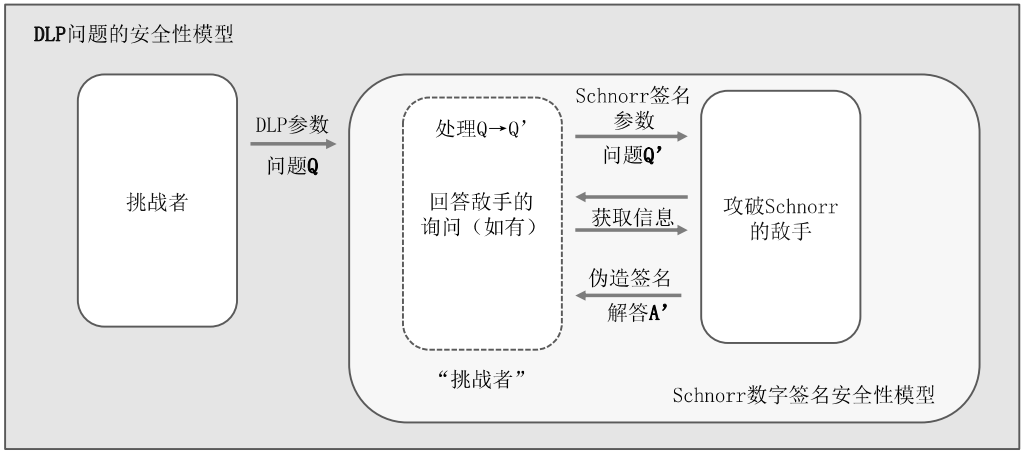
现代密码学三大法则：安全性证明



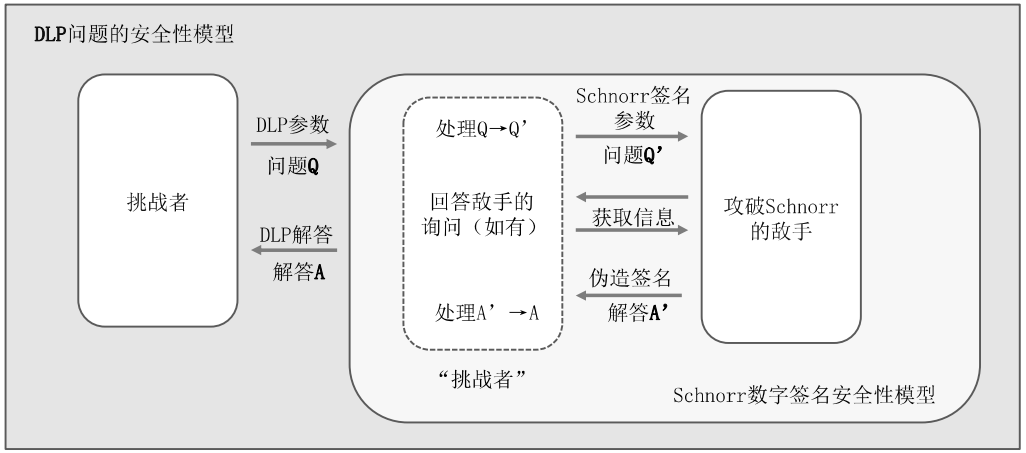
现代密码学三大法则：安全性证明



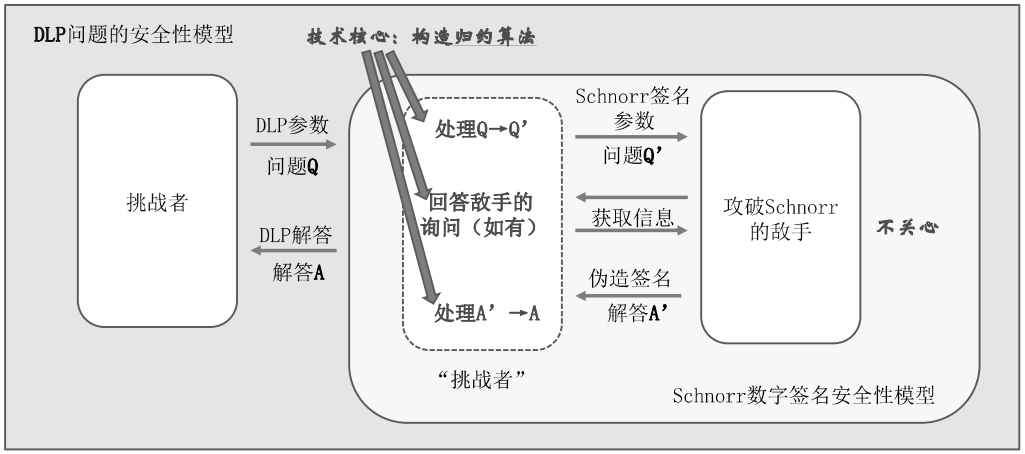
现代密码学三大法则：安全性证明



现代密码学三大法则：安全性证明



现代密码学三大法则：安全性证明

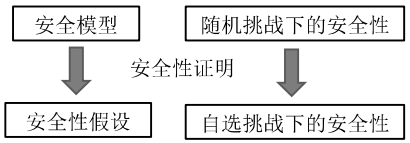


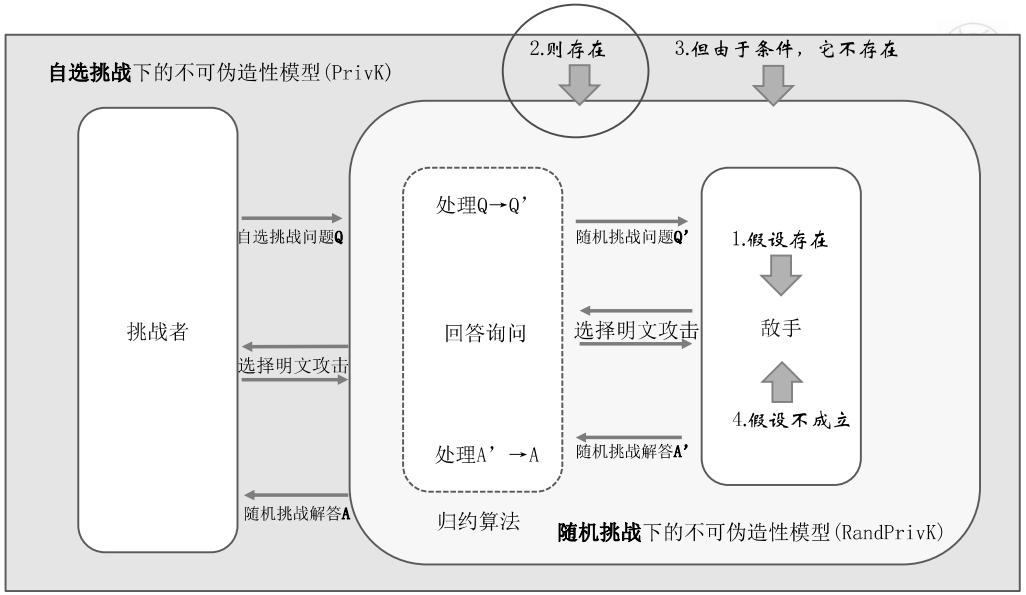
小试牛刀

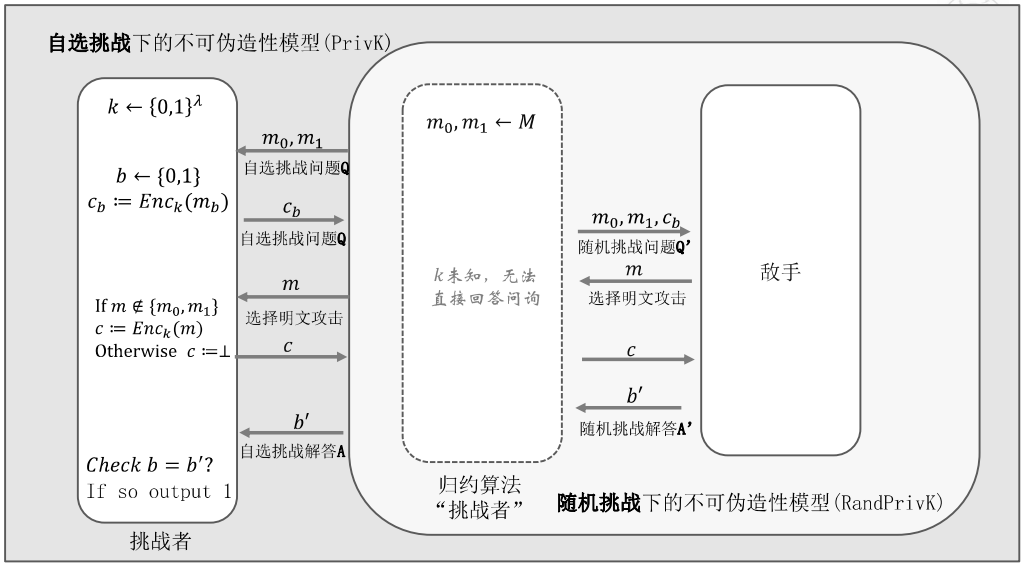
- 证明：在**选择明文攻击**下，如果一个私钥加密算法满足**自选挑战下**的不可区分性，则该算法也满足**随机挑战下**的不可区分性。

小试牛刀

- 证明：在**选择明文攻击**下，如果一个私钥加密算法满足**自选挑战下**的不可区分性，则该算法也满足**随机挑战下**的不可区分性。
- 思路：
  - 如果该算法不满足**随机挑战下**的不可区分性，则可以构造归约算法，攻击**自选挑战下**的不可区分性模型。
  - 由**安全性假设**得，可攻击**自选挑战下**不可区分性的敌手是不存在的，所以可以攻击**随机挑战下**的不可区分性的敌手是不存在的。

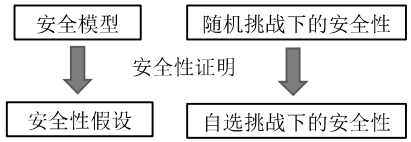






Formal Proof

1. Suppose a P.P.T. adversary $\mathcal{A}$ can break experiment $\boldsymbol{RandPrivK}^{\boldsymbol{CPA}}_{\boldsymbol{\Pi},\mathcal{A}}(\boldsymbol{\lambda})$ with **probability** $p(\lambda)$.
2. Then, we **construct a P.P.T. reduction** algorithm $\mathcal{R}^{\mathcal{A}}$ that can break $\boldsymbol{PrivK}^{\boldsymbol{CPA}}_{\boldsymbol{\Pi},\mathcal{R}}(\boldsymbol{\lambda})$ also with probability $p(\lambda)$.
3. However, since $\boldsymbol{\Pi}$ is secure under $\boldsymbol{PrivK}$ experiment, every P.P.T. algorithm (absolutely including $\mathcal{R}$) can break $\boldsymbol{PrivK}^{\boldsymbol{CPA}}_{\boldsymbol{\Pi},*}(\boldsymbol{\lambda})$ with at most **negligible** probability.
4. Thus, $p(\lambda)$ is a negligible function. In other words, **every** P.P.T adversary $\mathcal{A}$ can only break $\boldsymbol{RandPrivK}^{\boldsymbol{CPA}}_{\boldsymbol{\Pi},\mathcal{A}}(\boldsymbol{\lambda})$ with negligible probability. We complete the proof.



现代密码学三大法则

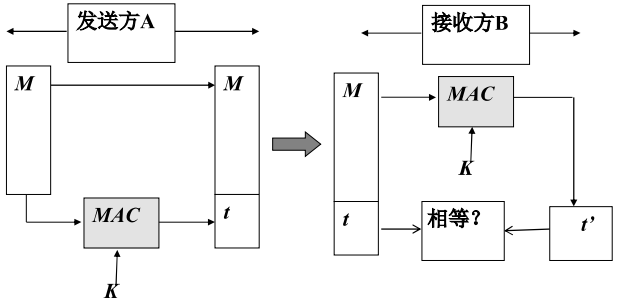
- Three main principles that distinguish modern cryptography from the classical cryptography
  - Principle 1 Formulation of Exact Definitions
  - Principle 2 Reliance on Precise Assumptions
  - Principle 3 Rigorous Proofs of Security

Constructions of Fixed-length MAC

袁泉

Message Authentication Codes 消息认证码

- Prevent an adversary **from modifying** a message sent by one party to another.



- Notice: Nothing can prevent an adversary from **impersonating** the party

Message Authentication Codes 消息认证码

DEFINITION 4.1 *A* message authentication code (*or* MAC) *consists of three probabilistic polynomial-time algorithms* (Gen, Mac, Vrfy) *such that:*

1. *The* key-generation algorithm Gen *takes as input the security parameter* 1^n *and outputs a key* k *with* $|k| \geq n$. — 密钥生成
2. *The* tag-generation algorithm Mac *takes as input a key* k *and a message* $m \in \{0,1\}^*$, *and outputs a tag* t. *Since this algorithm may be randomized, we write this as* $t \leftarrow \mathsf{Mac}_k(m)$. — 生成tag
3. *The deterministic* verification algorithm Vrfy *takes as input a key* k, *a message* m, *and a tag* t. *It outputs a bit* b, *with* $b = 1$ *meaning* valid *and* $b = 0$ *meaning* invalid. *We write this as* $b := \mathsf{Vrfy}_k(m,t)$. — 验证tag

Syntax 语法?

It is required that for every n, *every key* k *output by* $\mathsf{Gen}(1^n)$, *and every* $m \in \{0,1\}^*$, *it holds that* $\mathsf{Vrfy}_k(m, \mathsf{Mac}_k(m)) = 1$. — 完全性 Completeness

Security of MAC

■How to capture security?

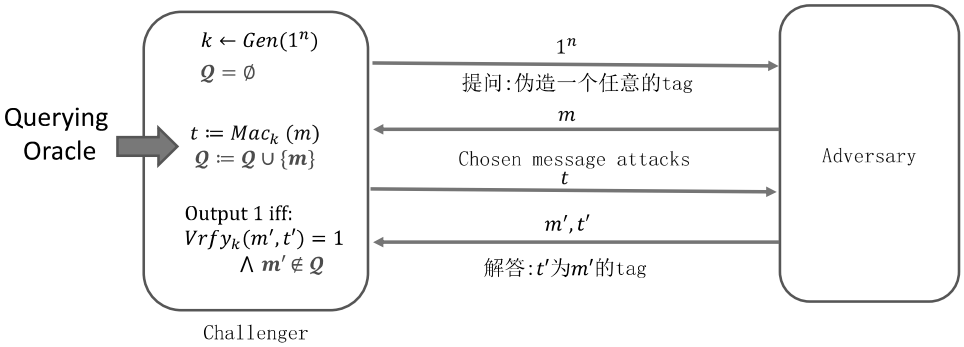
- 敌手计算能力 ⟵ Computational Security: P.P.T. adversary
- 敌手获取的信息 ⟵ Chosen Message Attacks: the adversary is able to obtain MAC tags on any messages it wishes during its Attack (Signing **Oracle**)
  选择消息攻击
- 敌手的目标 ⟵ Never chosen!

Generate a valid MAC tag on any "new" message

Security Models

■Existential unforgeability under (adaptive) chosen message attacks (EU-CMA):

选择消息攻击下的存在性不可伪造



Security Models

The message authentication experiment $\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)$**:**

1. *A key k is generated by running* $\mathsf{Gen}(1^n)$.
2. *The adversary $\mathcal{A}$ is given input 1^n and oracle access to* $\mathsf{Mac}_k(\cdot)$. *The adversary eventually outputs (m, t). Let $\mathcal{Q}$ denote the set of all queries that $\mathcal{A}$ asked its oracle.*
3. *$\mathcal{A}$* succeeds *if and only if (1)* $\mathsf{Vrfy}_k(m, t) = 1$ *and (2) $m \notin \mathcal{Q}$. In that case the output of the experiment is defined to be 1.*

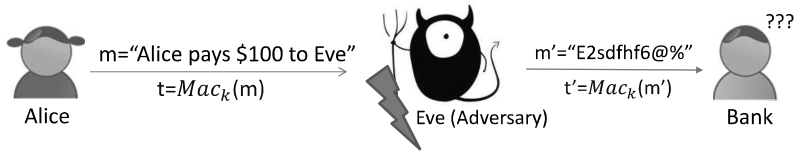
Security Models

DEFINITION 4.2 *A message authentication code* $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *is* existentially unforgeable under an adaptive chosen-message attack, *or just* secure, *if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there is a negligible function* negl *such that:*

$$\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

■Too strong?

➢Only authenticate "meaningful" message?



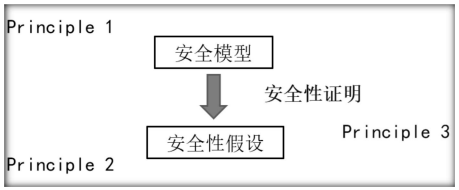
单选题 5分 设置

回顾：以上关于消息认证码（MAC）的知识对应了现代密码学三大法则中的哪一条？

A Formulation of Exact Definitions

B Reliance on Precise Assumptions

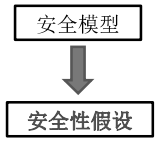
C Rigorous Proofs of Security



提交

Constructions of MAC

■ MAC的构造一般基于什么安全性假设？



■ 不同的数字签名可归约到不同的安全性假设

■ 安全性假设多种多样：

➢ 经典的数学困难问题

DLP问题，RSA问题（经典密码学） ⇐ 公钥密码

SIS问题，LWE问题（后量子密码学）

➢ 其他基础密码原语 ⇐ 公钥密码& 私钥密码

抗碰撞Hash函数，单向函数，伪随机函数……

➢ 非典型性的困难问题 & 多种安全性假设的混合

具体实例具体分析

Random Functions 随机函数

■ What is random function?

■ Consider $\mathbf{Func_n} = \{f(\cdot)\colon \{0,1\}^n \mapsto \{0,1\}^n\}$ be the set of functions that maps n-bit string to n-bit string.

We say $f \leftarrow \mathrm{Func_n}$ be a random function if f is **uniformly randomly** chosen from set $\mathrm{Func_n}$.

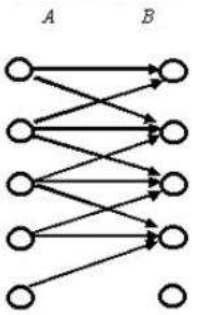
Random Functions 随机函数

■ What is the size of $\mathrm{Func_n} = \{f(\cdot)\colon \{0,1\}^n \mapsto \{0,1\}^n\}$?

$$\underbrace{2^n \cdot 2^n \cdot \ldots \cdot 2^n}_{2^n \text{ times}} = 2^{n \cdot 2^n}$$

➢The probability of guessing the value of a random function on an unobserved point is

$\frac{1}{2^n}$ when the output length is n



Random Functions 随机函数

■ A random function is **rarely** used.

■ Recording a random function mapping from $\{0,1\}^n$ to $\{0,1\}^n$ requires **at least** $n \cdot 2^n$ bits!

x	f(x)
00000000	01001000
00000001	11010101
00000010	11001010
.....	.....
11111111	01011100

Pseudorandom Functions 伪随机函数

■ **Looks like** a random function.

■Pseudorandom function (Syntax)

➢A keyed function

$$F_k : \{0,1\}^* \rightarrow \{0,1\}^*$$

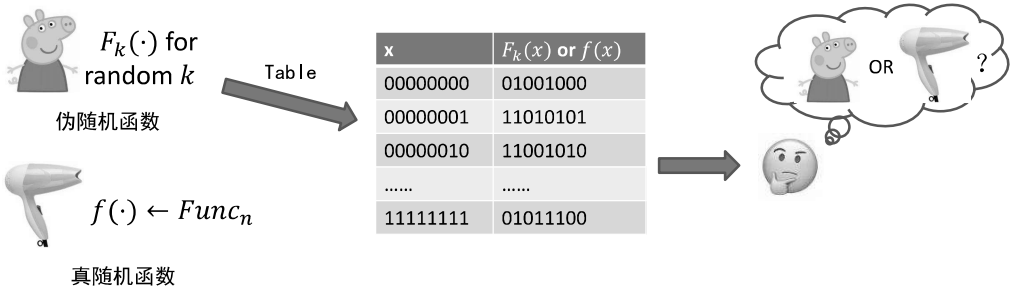
不可区分性！

➢F is pseudorandom if the function F_k (for **randomly-chosen** key k) is **indistinguishable** from a function chosen uniformly at random from the set of all functions having the same domain and range

$$F_k(x) \stackrel{\text{def}}{=} F(k,x) \qquad |F_k(x)| = |x| = |k|.$$

■ The storage of PRF only requires ***k*** and the **codes** of F.

Pseudorandom Functions 伪随机函数

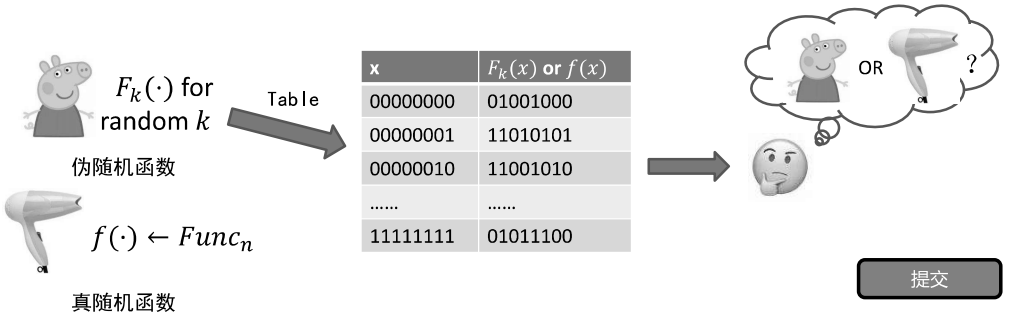


单选题 3分 设置

敌手是否应该知晓伪随机函数 F 的运行原理（代码）？

A 应该

B 不应该



提交

Pseudorandom Functions 伪随机函数

■敌手是否应该知晓伪随机函数 F 的运行原理？

应该！

Kerckhoffs' principle

对敌手（攻击者）的行为进行清晰的定义：
1. Kerckhoffs原则

■ Keys and Kerckhoffs' principle

➢19世纪末 , Auguste Kerckhoffs

- “The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.”

$F_k(\cdot)$

暴露　隐藏

■安全性必须仅依赖于密钥的隐私性，而不能依赖于算法的隐私性

Pseudorandom Functions 伪随机函数

对敌手（攻击者）的行为进行清晰的定义：
1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息
4. 计算能力

■如果敌手计算能力无限，能否进行暴力破解？

可以！

1. 已知F的构造，穷举所有$k \in \{0,1\}^n$，计算所有$F_k(\cdot)$的映射列表
2. 逐个与给定的映射列表（挑战）作比较，如果存在某个k使得$F_k(\cdot)$与挑战完全相同，则判断该函数为伪随机函数，否则为真随机函数

x	$F_k(\cdot)$ or $f(\cdot)$
00000000	01001000
00000001	11010101
00000010	11001010
.....	.....
11111111	01011100

思考题：该破解方法的成功率为多少？

Pseudorandom Functions 伪随机函数

■ 伪随机函数至多只能满足计算安全性，即针对多项式敌手的安全

■ 又有问题！

多项式敌手无法读取指数级大小的映射列表！

■ Recording a random function mapping from $\{0,1\}^n$ to $\{0,1\}^n$ requires **at least** $n \cdot 2^n$ bits!

对敌手（攻击者）的行为进行清晰的定义：
1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息
4. 计算能力

How to fix it?

x	$F_k(x)$ or $f(x)$
00000000	01001000
00000001	11010101
00000010	11001010
.....	.....
11111111	01011100

Pseudorandom Functions 伪随机函数

$F_k(\cdot)$ for random k

伪随机函数

Table

$f(\cdot) \leftarrow Func_n$

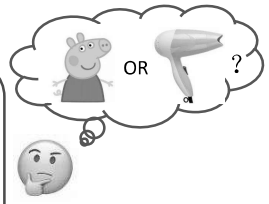
真随机函数

x	$F_k(\cdot)$ or $f(\cdot)$
00000000	01001000
00000001	11010101
00000010	11001010
.....	.....
11111111	01011100

x

$F_k(x)$ or $f(x)$

Polynomial times



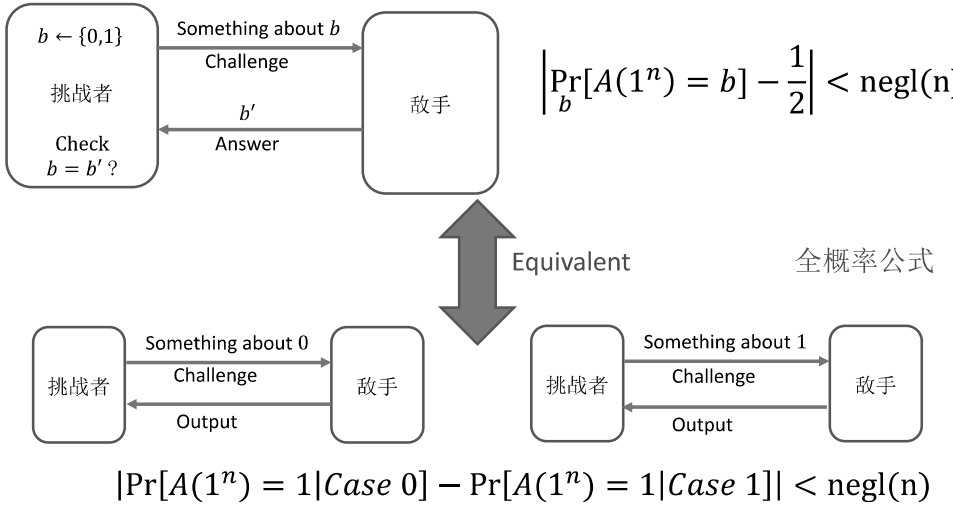
Pseudorandom Functions 伪随机函数

DEFINITION 3.25 *Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. F is a* pseudorandom function *if for all probabilistic polynomial-time distinguishers D, there is a negligible function* negl *such that:*

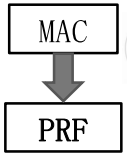
$$\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \leq \mathsf{negl}(n),$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $f \in \mathsf{Func}_n$ and the randomness of D.

Indistinguishability 不可区分性的定义



Construction of MAC based on PRF （※！）



CONSTRUCTION 4.5

Let F be a pseudorandom function. Define a fixed-length MAC for messages of length n as follows:

- Mac: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, output the tag $t := F_k(m)$. (If $|m| \neq |k|$ then output nothing.)
- Vrfy: on input a key $k \in \{0,1\}^n$, a message $m \in \{0,1\}^n$, and a tag $t \in \{0,1\}^n$, output 1 if and only if $t \stackrel{?}{=} F_k(m)$. (If $|m| \neq |k|$, then output 0.)

A fixed-length MAC from any pseudorandom function.

Construction of MAC based on PRF （※！）

THEOREM 4.6 *If F is a pseudorandom function, then Construction 4.5 is a secure fixed-length MAC for messages of length n.*

CONSTRUCTION 4.5

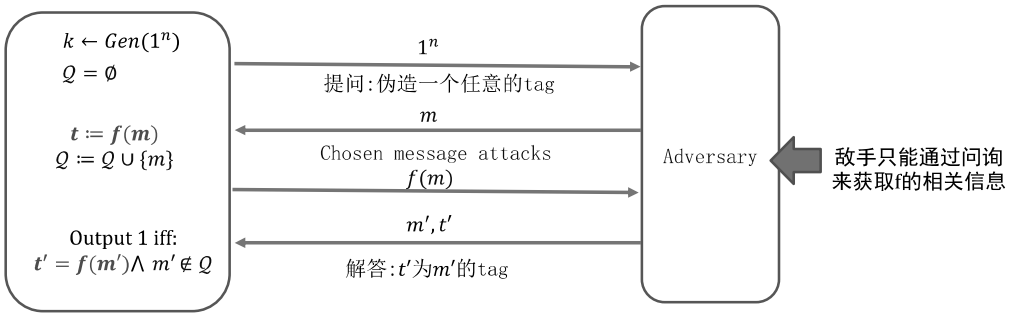
Let F be a pseudorandom function. Define a fixed-length MAC for messages of length n as follows:

- Mac: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, output the tag $t := F_k(m)$. (If $|m| \neq |k|$ then output nothing.)
- Vrfy: on input a key $k \in \{0,1\}^n$, a message $m \in \{0,1\}^n$, and a tag $t \in \{0,1\}^n$, output 1 if and only if $t \stackrel{?}{=} F_k(m)$. (If $|m| \neq |k|$, then output 0.)

Security Proof

Π: $Gen: k \leftarrow \{0,1\}^n$, $Mac_k(m) = F_k(m)$, $Vrfy_k(m,t): t = F_k(m)?$,

$\widetilde{\Pi}$: $\widetilde{Gen}: f \leftarrow Func_n$, $\widetilde{Mac_f}(m) = f(m)$, $\widetilde{Vrfy_f}(m,t): t = f(m)?$,



Attack the security of $\widetilde{\Pi}$ ⇔ Compute $f(m')$ for some **unqueried** m'

Security Proof

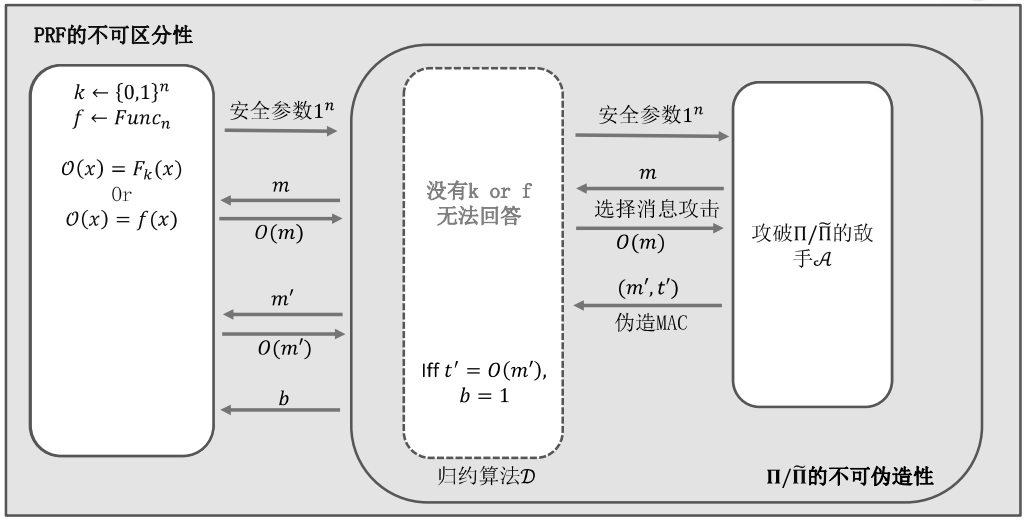
Let $\mathcal{A}$ be a probabilistic polynomial-time adversary. Consider the message authentication code $\widetilde{\Pi} = (\widetilde{\mathsf{Gen}}, \widetilde{\mathsf{Mac}}, \widetilde{\mathsf{Vrfy}})$ which is the same as $\Pi = (\mathsf{Mac}, \mathsf{Vrfy})$ in Construction 4.5 except that a truly random function f is used instead of the pseudorandom function F_k. That is, $\widetilde{\mathsf{Gen}}(1^n)$ works by choosing a uniform function $f \in \mathsf{Func}_n$, and $\widetilde{\mathsf{Mac}}$ computes a tag just as Mac does except that f is used instead of F_k. It is immediate that

$$\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\widetilde{\Pi}}(n) = 1] \leq 2^{-n} \tag{4.1}$$

because for any message $m \notin \mathcal{Q}$, the value $t = f(m)$ is uniformly distributed in $\{0,1\}^n$ from the point of view of the adversary $\mathcal{A}$.

Even for an adversary with unlimited computations!

Security Proof



Notice that if D's oracle is a pseudorandom function, then the view of $\mathcal{A}$ when run as a sub-routine by D is distributed identically to the view of $\mathcal{A}$ in experiment $\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)$. Furthermore, D outputs 1 exactly when $\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1$. Therefore

$$\Pr\left[D^{F_k(\cdot)}(1^n) = 1\right] = \Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1\right],$$

where $k \in \{0,1\}^n$ is chosen uniformly in the above. If D's oracle is a random function, then the view of $\mathcal{A}$ when run as a sub-routine by D is distributed identically to the view of $\mathcal{A}$ in experiment $\mathsf{Mac\text{-}forge}_{\mathcal{A},\widetilde{\Pi}}(n)$, and again D outputs 1 exactly when $\mathsf{Mac\text{-}forge}_{\mathcal{A},\widetilde{\Pi}}(n) = 1$. Thus,

$$\Pr\left[D^{f(\cdot)}(1^n) = 1\right] = \Pr\left[\mathsf{Mac\text{-}forge}_{\mathcal{A},\widetilde{\Pi}}(n) = 1\right],$$

where $f \in \mathsf{Func}_n$ is chosen uniformly.

Security Proof

Since F is a pseudorandom function and D runs in polynomial time, there exists a negligible function negl such that

$$\left|\Pr\left[D^{F_k(\cdot)}(1^n)=1\right]-\Pr\left[D^{f(\cdot)}(1^n)=1\right]\right|\leq \mathsf{negl}(n).$$

Assumption of PRF

Thus, for every P.P.T. adversary $\mathcal{A}$, it holds that

$$\left|\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1]-\Pr[\text{Mac-forge}_{\mathcal{A},\tilde{\Pi}}(n)=1]\right|\leq \mathsf{negl}(n);$$

Moreover, since it has been proven that

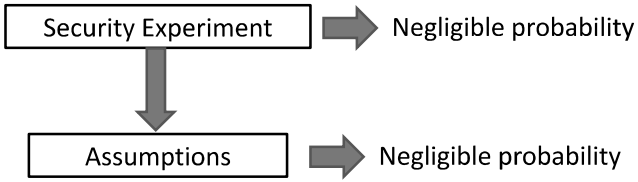
$$\Pr[\text{Mac-forge}_{\mathcal{A},\tilde{\Pi}}(n)=1]\leq 2^{-n}$$

we have

$$\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1]\leq 2^{-n}+\mathsf{negl}(n)$$

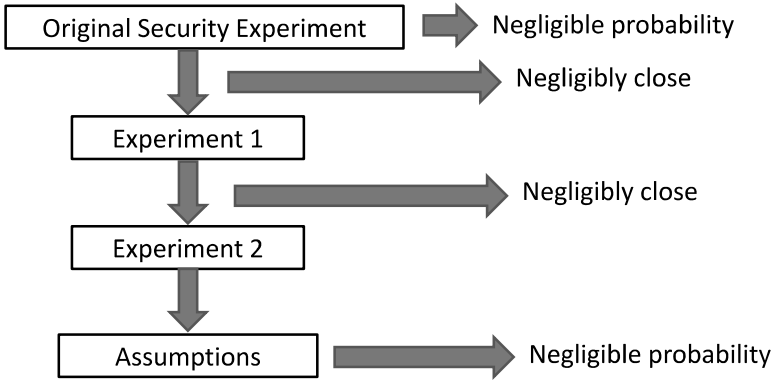
Security Proofs without hybrid argument

■Directly reduce the security to the assumptions.



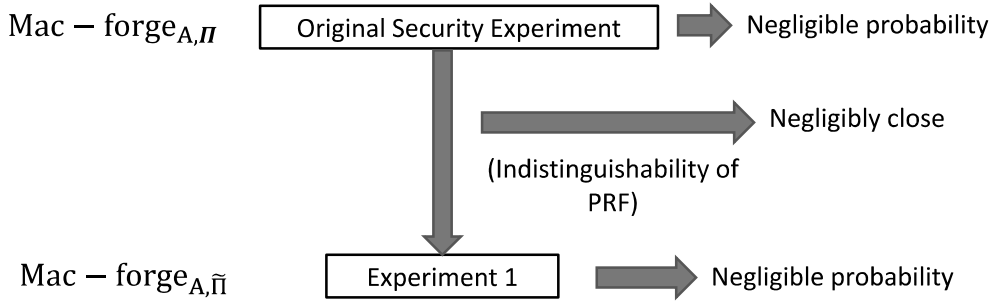
Security Proofs **with hybrid argument**

■Complete the security proof **by steps**.



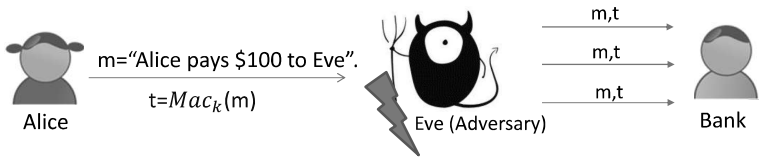
Security Proofs **with hybrid argument**

■Complete the security proof **by steps**.



Attack beyond Giving a Forgery

- Replay attacks
  - Example: Bank transfer
    - the MAC mechanism within itself does not prevent it.
- Preventing replay
  - Sequence numbers
  - Time stamp



DIGITAL SIGNATURES AND AUTHENTICATION

MAC for messages of arbitrary length

Homework

- Exercise 4.2, Page 148

4.2 Consider an extension of the definition of secure message authentication where the adversary is provided with both a Mac and a Vrfy oracle.

(a) Provide a formal definition of security for this case.

(b) Assume Π is a deterministic MAC using canonical verification that satisfies Definition 4.2. Prove that Π also satisfies your definition from part (a).

Submit to Canvas

Homework: Security Model with Vrfy Oracle

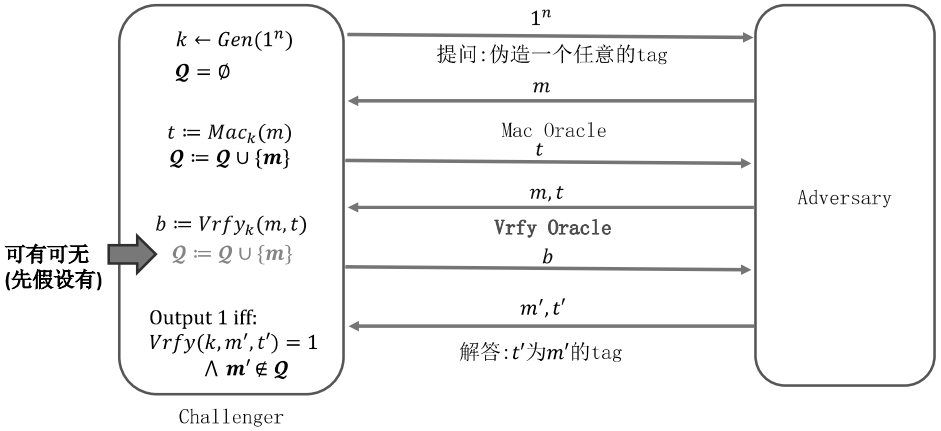
- Exercise 4.2, Page 148

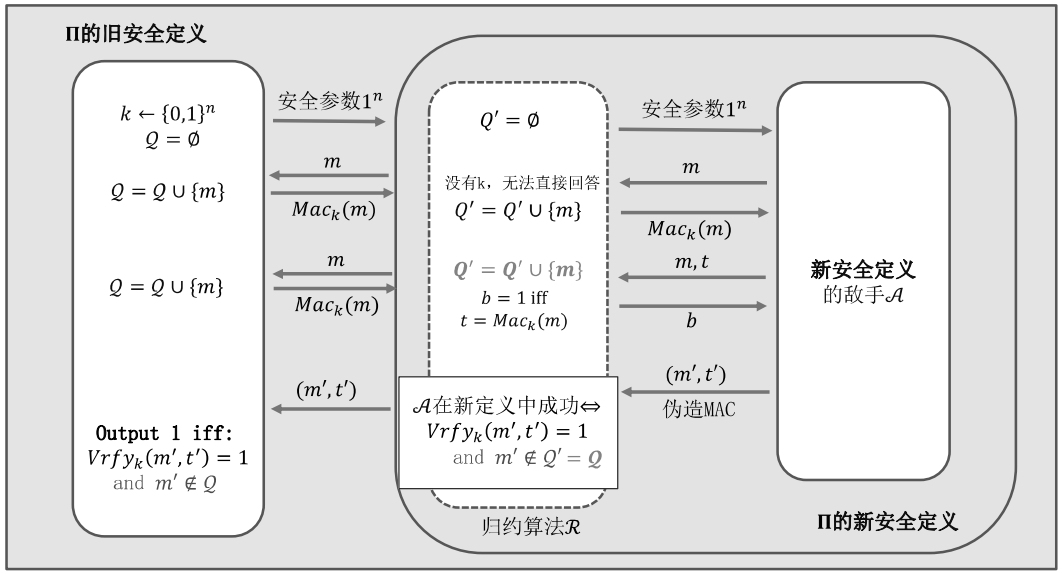
4.2 Consider an extension of the definition of secure message authentication where the adversary is provided with both a Mac and a Vrfy oracle.

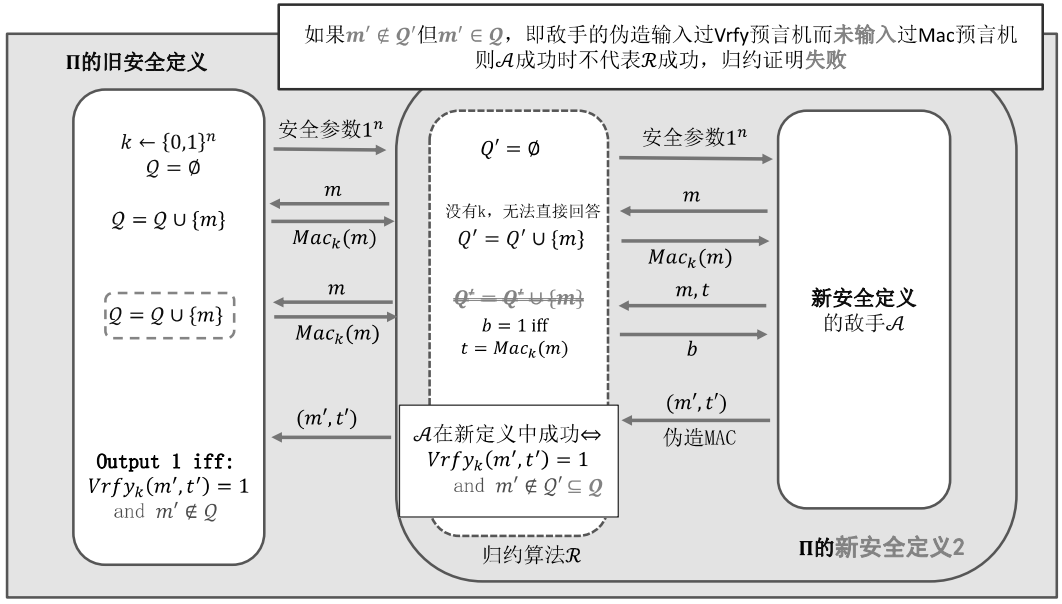
(a) Provide a formal definition of security for this case.

(b) Assume Π is a deterministic MAC using canonical verification that satisfies Definition 4.2. Prove that Π also satisfies your definition from part (a).

Homework: Security Model with Vrfy Oracle







CONSTRUCTION 4.5

Let F be a pseudorandom function. Define a fixed-length MAC for messages of length n as follows:

- Mac: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, output the tag $t := F_k(m)$. (If $|m| \neq |k|$ then output nothing.)
- Vrfy: on input a key $k \in \{0,1\}^n$, a message $m \in \{0,1\}^n$, and a tag $t \in \{0,1\}^n$, output 1 if and only if $t \stackrel{?}{=} F_k(m)$. (If $|m| \neq |k|$, then output 0.)

A fixed-length MAC from any pseudorandom function.

How to extend it to **arbitrary** length?

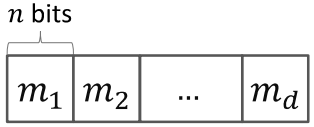
1汉字=2字节=16比特！

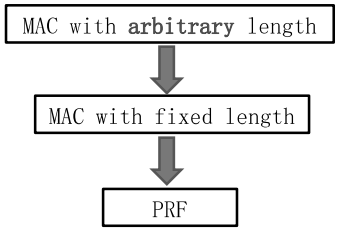
Domain Extension for MAC

Parse $m = m_1||m_2||\dots||m_d$

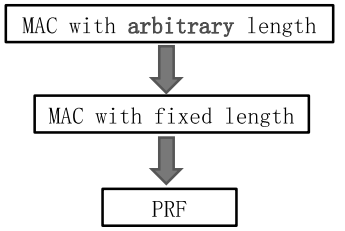
$d = \lceil |m|/n \rceil$: The number of **blocks**

The final block is padded with 0s if necessary

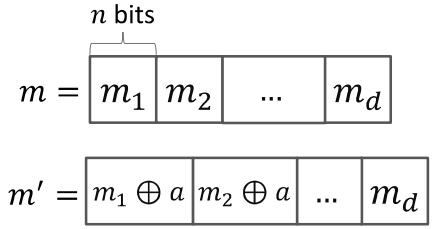




Domain Extension for MAC

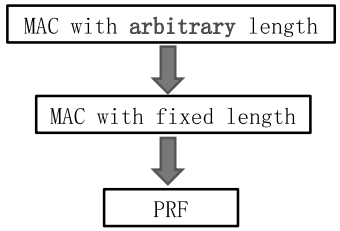


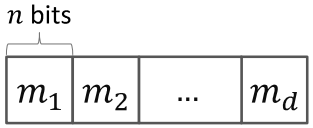
- XOR-and-MAC:
  - Given a fixed length Mac'
  - **Parse** $m = m_1||\dots||m_d$
  - $\tilde{m} = m_1 \oplus m_2 \oplus \cdots \oplus m_d \in \{0,1\}^n$
  - $t = Mac'_k(\tilde{m})$
- Attack:
  - Find $m' \neq m = m'_1||\dots||m'_d$
  - such that $m'_1 \oplus \cdots \oplus m'_d = \tilde{m}$
  - Forge $Mac_k(m') = Mac_k(m)$



Domain Extension for MAC

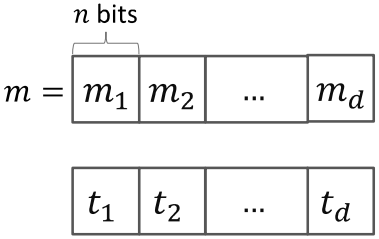
- XOR-and-MAC:
  - Given a fixed length Mac'
  - **Parse** $m = m_1||\dots||m_d$
  - $\tilde{m} = m_1 \oplus m_2 \oplus \cdots \oplus m_d \in \{0,1\}^n$
  - $t = Mac'_k(\tilde{m})$





Domain Extension for MAC

- MAC **separately**:
  - **Parse** $m = m_1||\dots||m_d$
  - $t_i = Mac'_k(m_i)$ for all $i \in [d]$
  - $t = t_1||\dots||t_d$



Domain Extension for MAC

- MAC **separately**:

  Parse $m = m_1 || \dots || m_d$

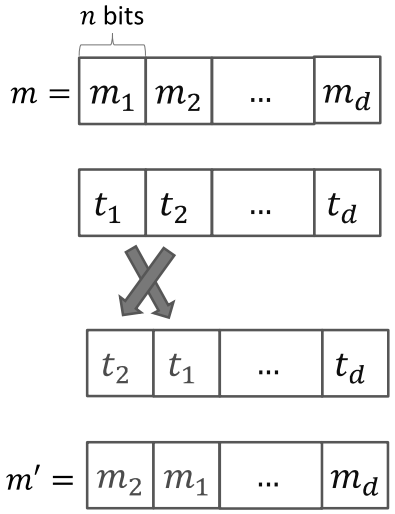
  $t_i = Mac'_k(m_i)$ for all $i \in [d]$

  $t = t_1 || \dots || t_d$

- Attack:

  Query $m = m_1 || m_2$

  Forge $t_2 || t_1$ for $m' = m_2 || m_1$



Domain Extension for MAC

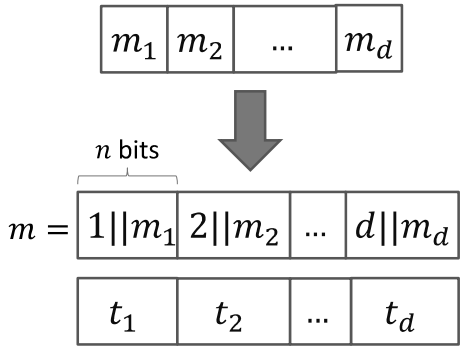
- MAC separately **with block index**:

  Parse $m = m_1 || \dots || m_d$

  The length of m_1 is shorter than n.

  $t_i = Mac'_k(\boldsymbol{i} || m_i)$ for all $i \in [d]$

  $t = t_1 || \dots || t_d$



Domain Extension for MAC

- MAC separately **with block index**:

  Parse $m = m_1 || \dots || m_d$

  The length of m_1 is shorter than n.

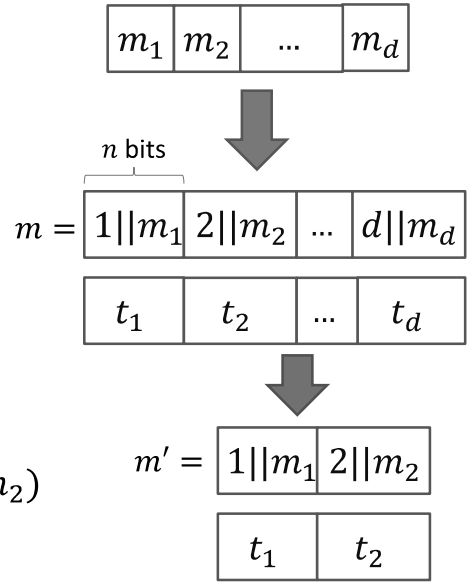
  $t_i = Mac'_k(\boldsymbol{i} || m_i)$ for all $i \in [d]$

  $t = t_1 || \dots || t_d$

- Attack:

  Truncation Attack. （**截断攻击**）

  $Mac_k(m_1 || \dots || m_d) \rightarrow Mac_k(m_1 || m_2)$



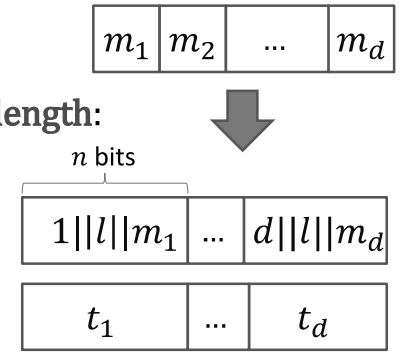
Domain Extension for MAC

- MAC separately with block index **and length**:

  Parse $m = m_1 || \dots || m_d$

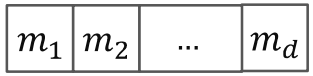
  $t_i = Mac'_k(i || \boldsymbol{l} || m_i)$ for all $i \in [d]$

  $t = t_1 || \dots || t_d$



思考题：为什么使用总长度 $\boldsymbol{l}$ 而不是总块数 $\boldsymbol{d}$?

Domain Extension for MAC

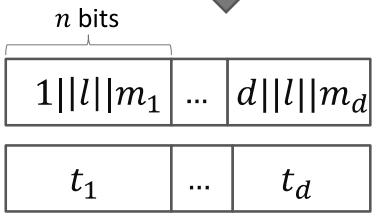


- MAC separately with block index **and length**:

  Parse $m = m_1 || \dots || m_d$

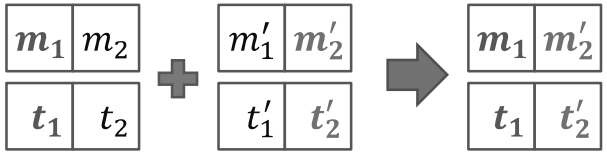
  $t_i = Mac'_k(i||l||m_i)$ for all $i \in [d]$

  $t = t_1 || \dots || t_d$



- Attack:

  Mix-and-match Attack.



Domain Extension for MAC

- MAC with random **identifier / salt / nonce.**

  Nonce: Number used **once.**

1. *The* key-generation algorithm Gen *takes as input the security parameter* 1^n *and outputs a key* k *with* $|k| \geq n$. 密钥生成
2. *The* tag-generation algorithm Mac *takes as input a key* k *and a message* $m \in \{0,1\}^*$*, and outputs a tag* t*. Since this algorithm may be randomized, we write this as* $t \leftarrow \mathsf{Mac}_k(m)$. 生成tag
3. *The deterministic* verification algorithm Vrfy *takes as input a key* k*, a message* m*, and a tag* t*. It outputs a bit* b*, with* $b = 1$ *meaning* valid *and* $b = 0$ *meaning* invalid*. We write this as* $b := \mathsf{Vrfy}_k(m,t)$. 验证tag

Syntax 语法?

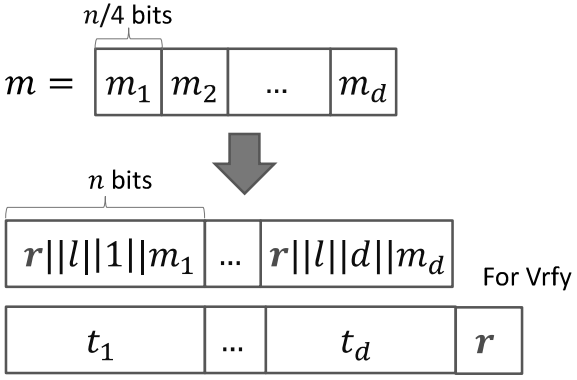
- Every time running Mac, independently generate a **random string.**

Domain Extension for MAC

- MAC with random **identifier / salt / nonce.**
- Every time running Mac, independently generate a **random string.**

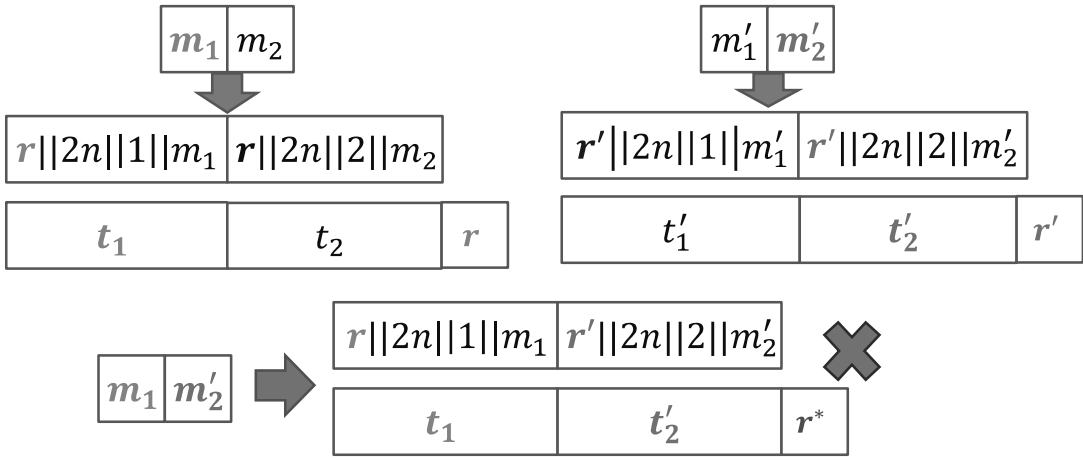
- $r \leftarrow \{0,1\}^{n/4}$
- $m_i, l \in \{0,1\}^{n/4}$

  Arbitrary length?



Domain Extension for MAC

- Mix-and-Match attack:



CONSTRUCTION 4.7

Let $\Pi' = (\mathsf{Mac}', \mathsf{Vrfy}')$ be a fixed-length MAC for messages of length n. Define a MAC as follows:

- Mac: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^*$ of (nonzero) length $\ell < 2^{n/4}$, parse m into d blocks $m_1, \ldots, m_d$, each of length $n/4$. (The final block is padded with 0s if necessary.) Choose a uniform identifier $r \in \{0,1\}^{n/4}$.

  For $i = 1, \ldots, d$, compute $t_i \leftarrow \mathsf{Mac}'_k(r \| \ell \| i \| m_i)$, where i, ℓ are encoded as strings of length $n/4$.† Output the tag $t := \langle r, t_1, \ldots, t_d \rangle$.
- Vrfy: on input a key $k \in \{0,1\}^n$, a message $m \in \{0,1\}^*$ of length $\ell < 2^{n/4}$, and a tag $t = \langle r, t_1, \ldots, t_{d'} \rangle$, parse m into d blocks $m_1, \ldots, m_d$, each of length $n/4$. (The final block is padded with 0s if necessary.) Output 1 if and only if $d' = d$ and $\mathsf{Vrfy}'_k(r \| \ell \| i \| m_i, t_i) = 1$ for $1 \le i \le d$.

† Note that i and ℓ can be encoded using $n/4$ bits because $i, \ell < 2^{n/4}$.

THEOREM 4.8 *If Π' is a secure fixed-length MAC for messages of length n, then Construction 4.7 is a secure MAC (for arbitrary-length messages).*

CONSTRUCTION 4.7

Let $\Pi' = (\mathsf{Mac}', \mathsf{Vrfy}')$ be a fixed-length MAC for messages of length n. Define a MAC as follows:

- Mac: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^*$ of (nonzero) length $\ell < 2^{n/4}$, parse m into d blocks $m_1, \ldots, m_d$, each of length $n/4$. (The final block is padded with 0s if necessary.) Choose a uniform identifier $r \in \{0,1\}^{n/4}$.

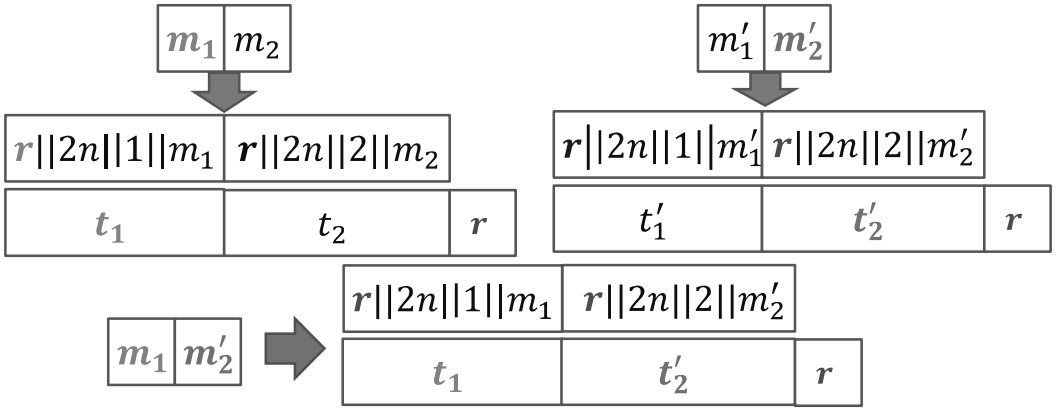
  For $i = 1, \ldots, d$, compute $t_i \leftarrow \mathsf{Mac}'_k(r \| \ell \| i \| m_i)$, where i, ℓ are encoded as strings of length $n/4$.† Output the tag $t := \langle r, t_1, \ldots, t_d \rangle$.
- Vrfy: on input a key $k \in \{0,1\}^n$, a message $m \in \{0,1\}^*$ of length $\ell < 2^{n/4}$, and a tag $t = \langle r, t_1, \ldots, t_{d'} \rangle$, parse m into d blocks $m_1, \ldots, m_d$, each of length $n/4$. (The final block is padded with 0s if necessary.) Output 1 if and only if $d' = d$ and $\mathsf{Vrfy}'_k(r \| \ell \| i \| m_i, t_i) = 1$ for $1 \le i \le d$.

† Note that i and ℓ can be encoded using $n/4$ bits because $i, \ell < 2^{n/4}$.

Security Proof

- Whether $\boldsymbol{r}$ is **repeated** in the signing queries?

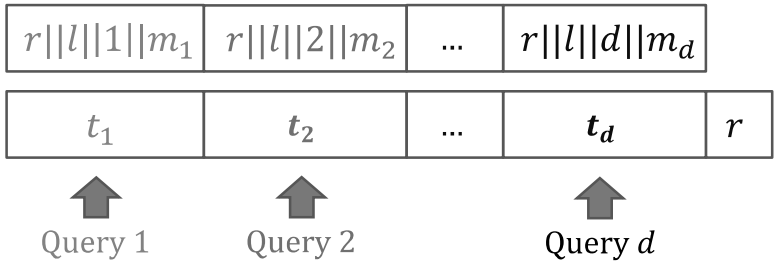
  If the random $\boldsymbol{r}$ is **repeated**, then the security is broken.



Security Proof

- Whether $\boldsymbol{r}$ is **repeated** in the signing queries?
- Is there a **new block** in the forgery?

  The forgery may be given by mix-and-match from signing queries.



Security Proof

- Whether $\boldsymbol{r}$ is **repeated** in the signing queries?
- Is there a **new block** in the forgery?

- **Repeat**: The event that there exists a random identifier $\boldsymbol{r}$ picked more than once in the signing queries.
- **NewBlock**: The event that the final forgery

$$(m', t') = \left(m'_1 || \dots || m'_{d'}, r' || t'_1 || \dots || t'_{d'}\right)$$

contains a block such that $(r'||l'||i||m'_i)$ is **never** computed by Mac'$(\cdot)$ in the signing queries.

Security Proof

We have

$$\begin{aligned}\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1] &= \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \text{Repeat}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\text{Repeat}} \wedge \text{NewBlock}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}] \\ &\leq \Pr[\text{Repeat}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \text{NewBlock}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}].\end{aligned}$$

Security Proof

We have

$$\begin{aligned}\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1] &= \boxed{\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \text{Repeat}]} \\ &\quad + \boxed{\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\text{Repeat}} \wedge \text{NewBlock}]} \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}] \\ &\leq \boxed{\Pr[\text{Repeat}]} \\ &\quad + \boxed{\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \text{NewBlock}]} \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}].\end{aligned}$$

Security Proof

- **Step 1:** Pr[Repeat] is negligible.

LEMMA A.15 *Fix a positive integer N, and say q elements $y_1, \dots, y_q$ are chosen uniformly and independently at random from a set of size N. Then the probability that* there exist distinct i, j with $y_i = y_j$ *is at most $\frac{q^2}{2N}$. That is,*

$$\text{coll}(q, N) \leq \frac{q^2}{2N}.$$

（生日攻击成功率）

$r_j \leftarrow \{0,1\}^{n/4}$ for $j \in \{1,2,\dots,q(n)\}$

Pr[Repeat] $\leq \frac{q(n)^2}{2^{n/4+1}}$, where $q(n)$ denotes the number of signing queries.

Security Proof

- **Step 2:** $\Pr[\text{Macforge}_{A,\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{Newblock}}]$

NewBlock: The event that the final forgery contains a block such that $(r'||l'||i||m_i')$ is **never** computed by Mac'$(\cdot)$ in the signing queries.

➢如果Repeat和NewBlock同时**未发生**，则敌手可能成功吗？

设敌手输出的伪造为 $m' = m_1'||\ldots||m_{d'}'$， $t' = r'||t_1'||\ldots||t_{d'}'$

签名预言机所选随机数分别是 $r_1, \ldots, r_q$且两两不相同

1. 若$r' \notin \{r_1, \ldots r_q\}$，则显然NewBlock已发生。

Security Proof

$\Pr[\text{Macforge}_{A,\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{Newblock}}] = 0$

NewBlock: The event that the final forgery contains a block such that $(r'||l'||i||m_i')$ is **never** computed by Mac'$(\cdot)$ in the signing queries.

➢如果Repeat和NewBlock同时**未发生**，则敌手可能成功吗？

2. 若 $r' = r_j$, 观察第 j 次的问询过程 $m = m_1||\ldots||m_d$

2.1 若$l' \neq l$， NewBlock已发生。

2.2 若$l' = l$，则$d' = d$

则每一个m_i'与m_i**都相等**

$m' = m$，$\boldsymbol{m', t'}$ **不是伪造**

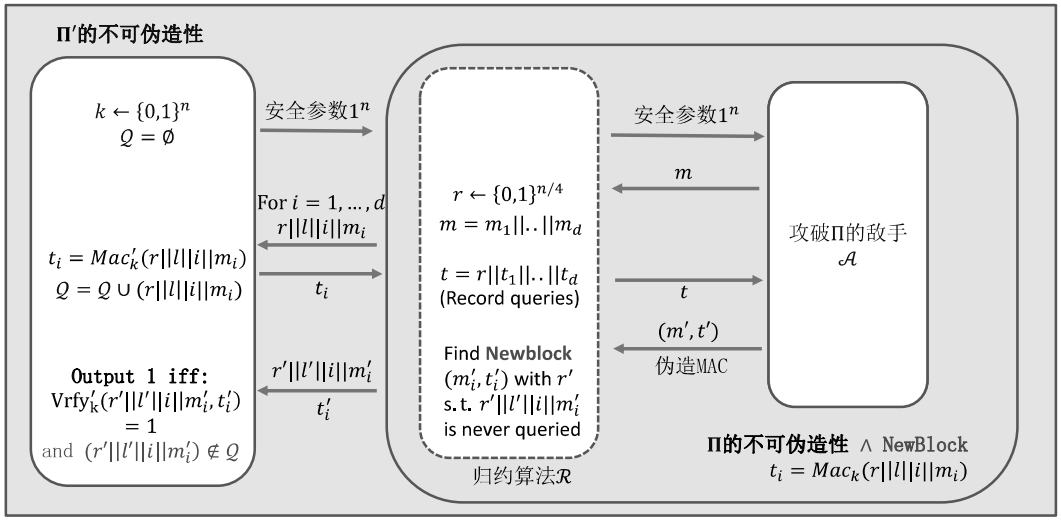
$r'\|\|l'\|\|1\|\|m_1'$	$r'\|\|l'\|\|2\|\|m_2'$	...	$r'\|\|l'\|\|d\|\|m_{d'}'$
$r_j\|\|l\|\|1\|\|m_1$	$r_j\|\|l\|\|\|\|m_2$	...	$r_j\|\|l\|\|d\|\|m_d$

Security Proof

- **Step 3:** $\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{NewBlock}]$ *is negligible.*
- 如果NewBlock发生了，则意味着敌手在伪造中包含了一个(m_i', t_i')，使得 t_i'是 $(r'||l'||i||m_i')$对Mac'的合法标签，且 $\text{Mac}_k'(r'||l'||i||m_i')$ **未在问询中被计算过**
- 如果存在P.P.T.的敌手A完成该事件的成功率为不可忽略的，则可构造**归约算法**攻击Mac'的安全性

Security Proof

（原课件有误）



Security Proof

1. Suppose a P.P.T. adversary $\mathcal{A}$ such that
$$\Pr\left[\text{Macforge}_{\Pi,\mathcal{A}}(n) = 1 \wedge \text{NewBlock}\right] = p(n).$$
2. Then, we construct a P.P.T. reduction algorithm $\mathcal{R}^{\mathcal{A}}$ that can break $\text{Macforge}_{\Pi',\mathcal{R}}(n)$ also with probability $p(n)$.
3. Since $\boldsymbol{\Pi}$ is secure, every P.P.T. algorithm can break $\text{Macforge}_{\Pi',\mathcal{R}}(n)$ with at most **negligible** probability.
4. Thus, for every P.P.T. adversary $\mathcal{A}$, there exists a negligible function $negl$ such that
$$\Pr\left[\text{Macforge}_{\boldsymbol{\Pi},\mathcal{A}}(n) = 1 \wedge \text{NewBlock}\right] = p(n) \leq negl(n).$$

Security Proof

We have

$$\begin{aligned}\Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1] &= \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{Repeat}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \text{NewBlock}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}] \\ &\leq \Pr[\text{Repeat}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{NewBlock}] \\ &\quad + \Pr[\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Repeat}} \wedge \overline{\text{NewBlock}}]. \\ &\leq \boldsymbol{q(n) \cdot 2^{-n/4-1} + negl(n).}\end{aligned}$$

单选题 1分 设置

此题未设置答案，请点击右侧设置按钮

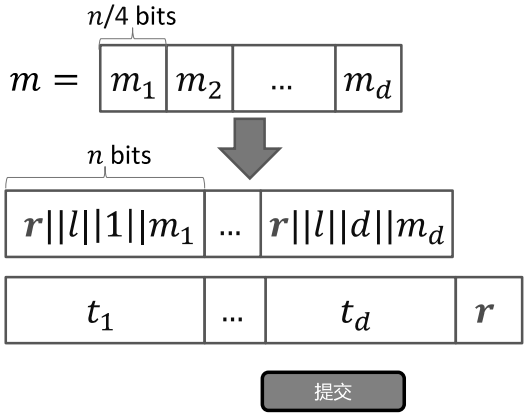
对于长度为 l 比特的消息 m (假设l为n的倍数), 该Mac的长度为?

A l 比特

B $l/4$ 比特

C $4l$ 比特

D $(4l + n/4)$ 比特



提交

Domain Extension for MAC

➢对于长度为 l 的消息 m, 该Mac的长度为?

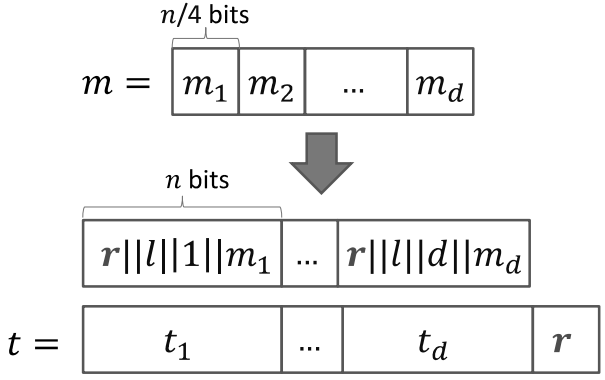
➢ $d = \left\lceil \frac{|m|}{n/4} \right\rceil \geq \frac{l}{n/4} = \frac{4l}{n}$

➢ $t = n * |t_i| + |r|$

$\geq \frac{4l}{n} * n + n/4$

$= 4l + n/4$

➢ **Too expensive!**

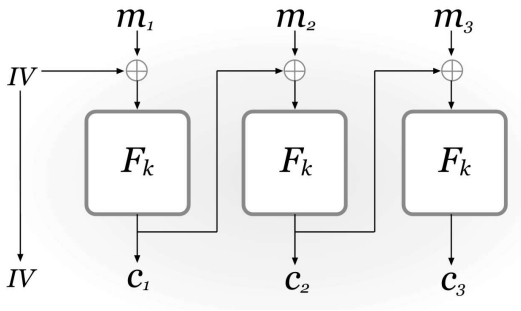


CBC-MAC

CBC-MAC

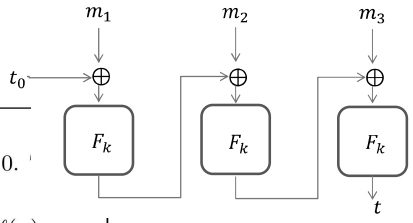
■CBC加密：分组加密 F_k: $\{0,1\}^n \rightarrow \{0,1\}^n$

Parse $m = m_1 || \ldots || m_x$, $m_i \in \{0,1\}^n$, Pick random $IV \leftarrow \{0,1\}^n$.



解密：$m_i = Dec_k(c_i) \oplus c_{i-1}$ (and $m_1 = Dec_k(c_1) \oplus IV$)

CBC-MAC



CBC-MAC for fixed length

CONSTRUCTION 4.11

Let F be a pseudorandom function, and fix a length function $\ell > 0$. The basic CBC-MAC construction is as follows:

- Mac: on input a key $k \in \{0,1\}^n$ and a message m of length $\ell(n) \cdot n$, do the following (we set $\ell = \ell(n)$ in what follows):
  1. Parse m as $m = m_1, \ldots, m_\ell$ where each m_i is of length n.
  2. Set $t_0 := 0^n$. Then, for $i = 1$ to ℓ:
     Set $t_i := F_k(t_{i-1} \oplus m_i)$.

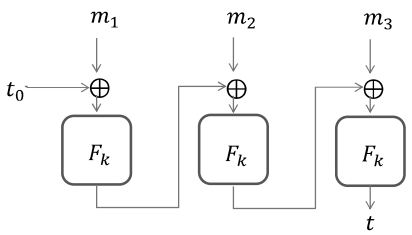
  Output t_ℓ as the tag.
- Vrfy: on input a key $k \in \{0,1\}^n$, a message m, and a tag t, do: If m is not of length $\ell(n) \cdot n$ then output 0. Otherwise, output 1 if and only if $t \stackrel{?}{=} \mathsf{Mac}_k(m)$.

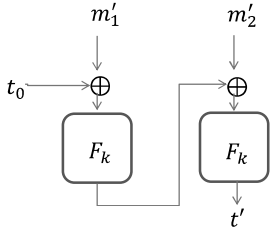
Basic CBC-MAC (for fixed-length messages).

CBC-MAC

THEOREM 4.12 *Let ℓ be a polynomial. If F is a pseudorandom function, then Construction 4.11 is a secure MAC for messages of length $\ell(n) \cdot n$.*

Not secure with varying length?

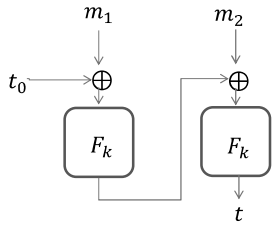




CBC-MAC

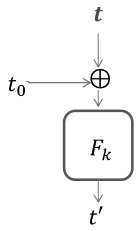
Forgery attack with varying length:

(1) Query $m_1||m_2$



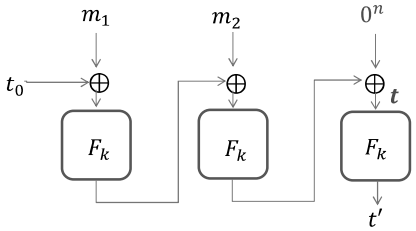
Obtain t

(2) Query $t \oplus t_0$



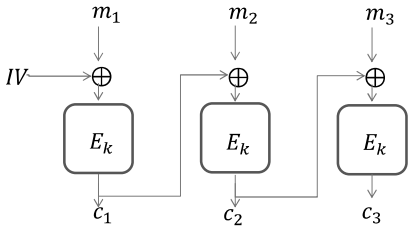
Obtain $t' = F_k(t)$

(3) Forgery $m_1||m_2||0^n, t'$

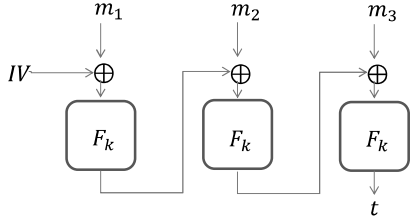


CBC-MAC vs CBC-Encryption

- CBC encryption uses a random IV
  - CBC-MAC uses fixed IV (t_0)
  - How about random IV for CBC-MAC?
- In CBC encryption all blocks are output by the encryption algorithm
  - CBC-MAC only output the final block
  - How about outputting all blocks for CBC-MAC?

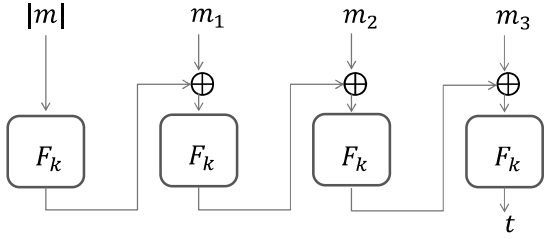


$c = (IV, c_1, c_2, c_3)$



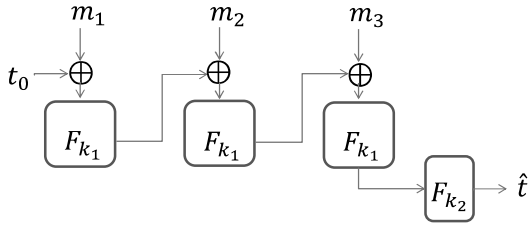
Secure CBC-MAC for variable-length messages

1. *Prepend* the message m with its length $|m|$ (encoded as an n-bit string), and then compute basic CBC-MAC on the result;

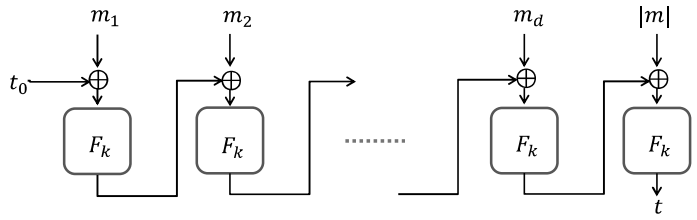


Secure CBC-MAC for variable-length messages

2. Change the scheme so that key generation chooses two independent, uniform keys $k_1 \in \{0,1\}^n$ and $k_2 \in \{0,1\}^n$. Then to authenticate a message m, first compute the basic CBC-MAC of m using k_1 and let t be the result; output the tag $\hat{t} := F_{k_2}(t)$.



How about appending the message with its block length **at the end**?

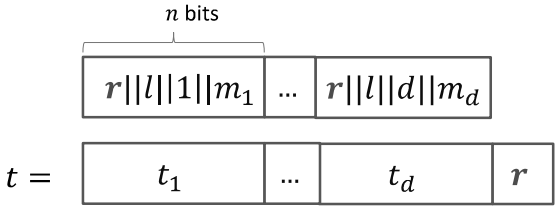


Hash Functions and HMAC

Domain Extension for MAC

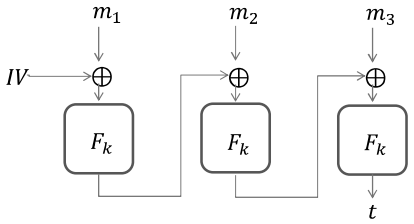
- Mac Separately:

  Fixed length → Arbitrary length



- CBC-MAC

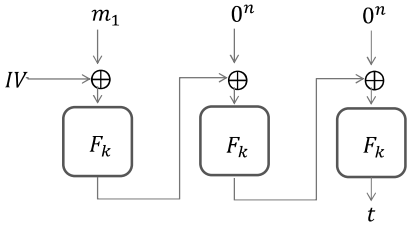
  PRF → Longer but **fixed** length



任意定长≠任意长！

Fixed length with long-enough bits

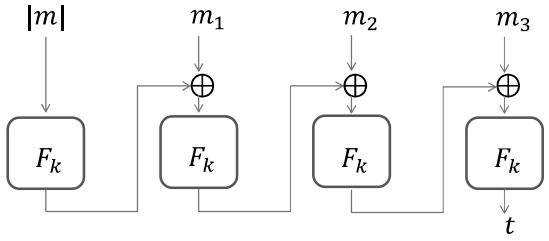
1. Given a shorter message m_1:
2. Extend it to $m = m_1||0^n||0^n$,
3. Generate CBC-MAC with length $3n$.



Waste of time! 2 meaningless computations of F_k.

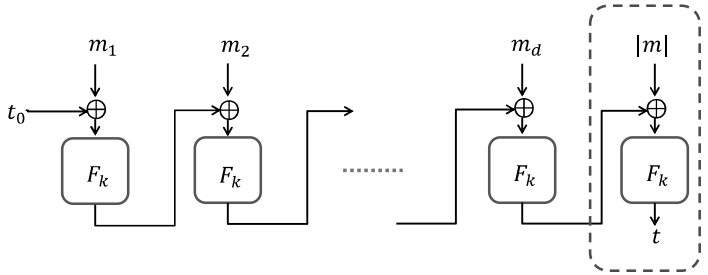
Secure CBC-MAC for variable-length messages

1. *Prepend* the message m with its length $|m|$ (encoded as an n-bit string), and then compute basic CBC-MAC on the result;



Secure CBC-MAC for variable-length messages

Note that appending $|m|$ to the *end* of the message and then computing the basic CBC-MAC is *not* secure.

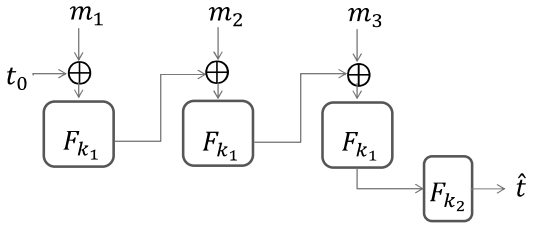


Hint: Query $m_1, m_2, (m_1||n||m_3)$ and obtain t_1, t_2, t_3,

then forge a tag for $(m_2||n||(t_1 \oplus t_2 \oplus m_3))$

Secure CBC-MAC for variable-length messages

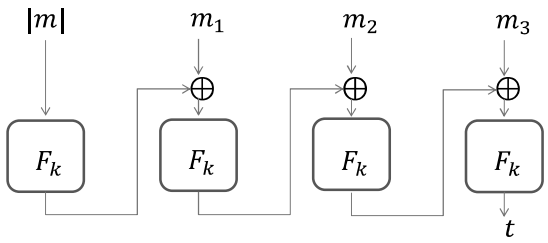
2. Change the scheme so that key generation chooses two independent, uniform keys $k_1 \in \{0,1\}^n$ and $k_2 \in \{0,1\}^n$. Then to authenticate a message m, first compute the basic CBC-MAC of m using k_1 and let t be the result; output the tag $\hat{t} := F_{k_2}(t)$.



Domain Extension for MAC

➢ Improved CBC-MAC

PRF → **Arbitrary** length



- Tag size: Polynomial in n and **independent in $|m|$.**
- Running time: **Linear in $|m|$**

Hash Functions

Syntax of Hash Functions

基本定义

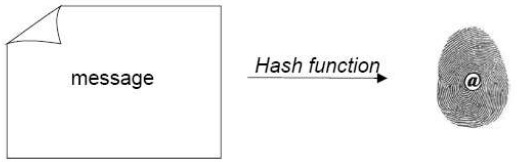
$y = h(x)$

- 将任意长度的消息x压缩为固定长度的杂凑值y，即

$$h: \{0,1\}^* \to \{0,1\}^n,$$

$\{0,1\}^*$：任意长度消息的集合
$\{0,1\}^n$：长度为n的比特串的集合

- 通常使用的Hash函数都是公开的



压缩性！
|定义域|>|值域|

Syntax of Hash Functions

DEFINITION 5.1 *A* hash function *(with output length ℓ) is a pair of probabilistic polynomial-time algorithms* (Gen, H) *satisfying the following:*

- Gen *is a probabilistic algorithm which takes as input a security parameter* 1^n *and outputs a key* s. *We assume that* 1^n *is implicit in* s.
- H *takes as input a key* s *and a string* $x \in \{0,1\}^*$ *and outputs a string* $H^s(x) \in \{0,1\}^{\ell(n)}$ *(where* n *is the value of the security parameter implicit in* s*).*

If H^s *is defined only for inputs* $x \in \{0,1\}^{\ell'(n)}$ *and* $\ell'(n) > \ell(n)$*, then we say that* (Gen, H) *is a* fixed-length hash function for inputs of length ℓ'. *In this case, we also call* H *a* compression function.

Syntax of Hash Functions

Formally, we consider *keyed* hash functions. $H^s(x) \stackrel{\text{def}}{=} H(s,x)$.

1. 哈希函数的密钥不一定是**均匀随机的**字符串。

$$s \leftarrow Gen(1^n)$$

2. 哈希函数的密钥常常是**公开的**。

Security Model

对敌手（攻击者）的行为进行清晰的定义：
1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息
4. 计算能力

How to define a secure hash function?

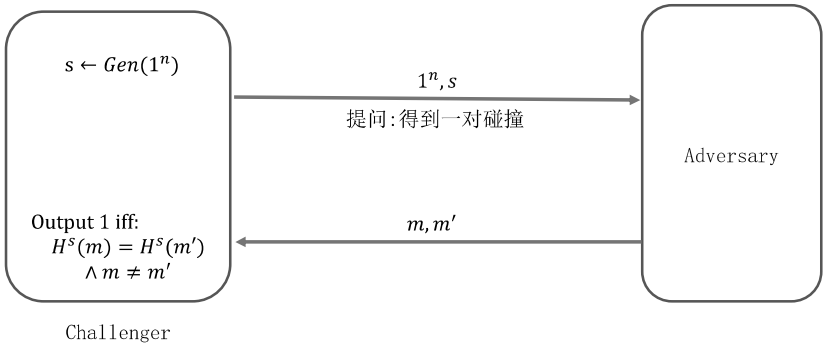
1. **抗碰撞性 Collision resistance (CRH)**
2. 抗原像性 Preimage resistance
3. 抗第二原像性 Second-preimage resistance

And more….

(*weak* collision resistance, collapsingness, decisional second-preimage resistance, multi-target second-preimage resistance…)

Security Model

The collision-finding experiment Hash-coll$_{\mathcal{A},\Pi}(n)$**:**



单选题 7分 设置

在对哈希函数安全模型的定义中，对敌手计算能力的要求应为：

对敌手（攻击者）的行为进行清晰的定义：
1. Kerckhoffs原则
2. 攻击目标
3. 能获取的信息
4. 计算能力

A 多项式时间的

B 指数时间的

C 无限时间的

提交

Syntax of Hash Functions

基本定义

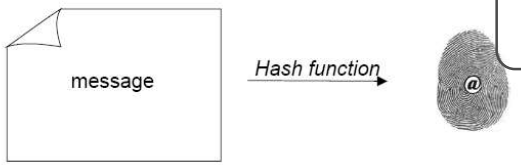
$y = h(x)$

- 将任意长度的消息x压缩为固定长度的杂凑值y，即

$$h: \{0,1\}^* \to \{0,1\}^n,$$

$\{0,1\}^*$：任意长度消息的集合
$\{0,1\}^n$：长度为n的比特串的集合

- 通常使用的Hash函数都是公开的

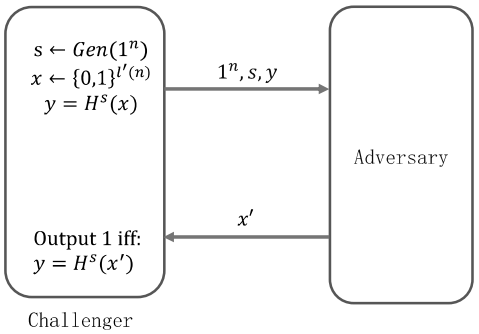


压缩性！
|定义域|>|值域|

Security Models（定义不唯一）

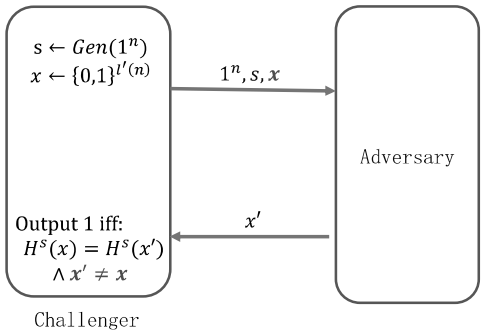
抗原像性 Preimage resistance

单向性 One-wayness



抗第二原像性 Second-preimage resistance

抗目标碰撞性 Target collision resistance



Security Models

抗碰撞性 ⇒ 抗第二原像性 ⇏ 抗原像性

Hint: $h(x) = \begin{cases} 0||x & if\ x < 2^n \\ 1||g(x) & if\ x \geq 2^n \end{cases}$

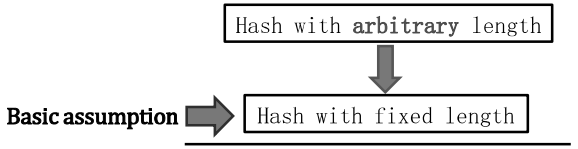
$g(x)$ is a collision-resistant hash function

（课后习题5.1题目有误）

How to construct secure hash functions:
The Merkle-Damgård Transform

How to construct hash functions?

- Assume (Gen, h) be a collision-resistant hash function mapping 2n-bit string to n-bit string.
- Construct (Gen, H) mapping $x \in \{0,1\}^*$ to n-bit string.



(Symmetric Hash Functions)

The Merkle-Damgård Transform

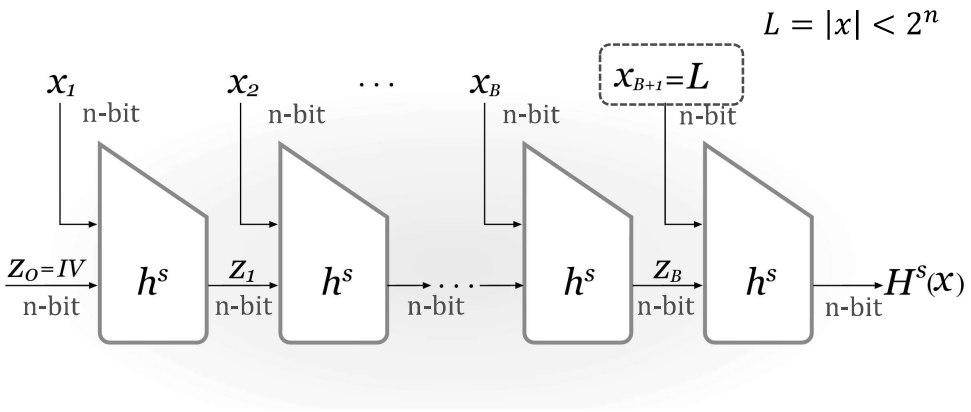


FIGURE 5.1: The Merkle–Damgård transform.

The Merkle-Damgård Transform

CONSTRUCTION 5.3

Let (Gen, h) be a fixed-length hash function for inputs of length $2n$ and with output length n. Construct hash function (Gen, H) as follows:

- Gen: remains unchanged.
- H: on input a key s and a string $x \in \{0,1\}^*$ of length $L < 2^n$, do the following:
  1. Set $B := \lceil \frac{L}{n} \rceil$ (i.e., the number of blocks in x). Pad x with zeros so its length is a multiple of n. Parse the padded result as the sequence of n-bit blocks $x_1, \ldots, x_B$. Set $x_{B+1} := L$, where L is encoded as an n-bit string.
  2. Set $z_0 := 0^n$. (This is also called the IV.)
  3. For $i = 1, \ldots, B+1$, compute $z_i := h^s(z_{i-1} \| x_i)$.
  4. Output z_{B+1}.

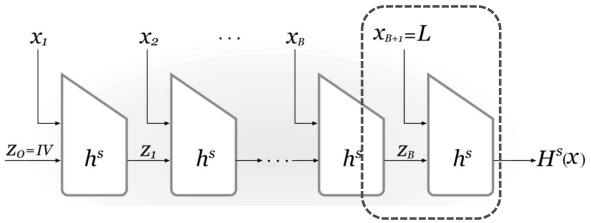
The Merkle-Damgård Transform

THEOREM 5.4 *If* (Gen, h) *is collision resistant, then so is* (Gen, H).

- Assume $\mathcal{A}$ finding a collision x and x' such that $H^s(x) = H^s(x')$, we construct a reduction $\mathcal{R}$ finding a collision of h^s:
- Case 1: $|x| \neq |x'|$ $(L \neq L')$

  $Z_B||L \neq Z_{B'}||L'$

  Thus, $Z_B||L$ and $Z_{B'}||L'$ is a collision of h^s



The Merkle-Damgård Transform

$x = |x_1|x_2|x_3|x_4|$

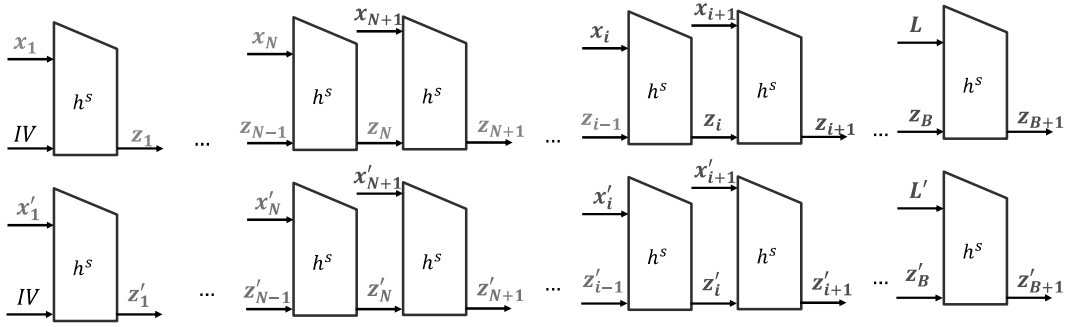
$x' = |x'_1|x'_2|x'_3|x'_4|$

$(N = 2)$

- Case 2: $|x| = |x'|$ $(L = L'$ and thus $B = B')$

  Let N ∈ {1, … , B} be the **largest** index such that $x_N \neq x'_N$

  There must be $\boldsymbol{i} \geq N$ such that $z_{i-1} \neq z'_{i-1}$ but $z_i = z'_i$



单选题 3分 设置

Case 2中归约算法应输出什么作为h^s的碰撞?

A x_N 与 x'_N

C x_i 与 x'_i

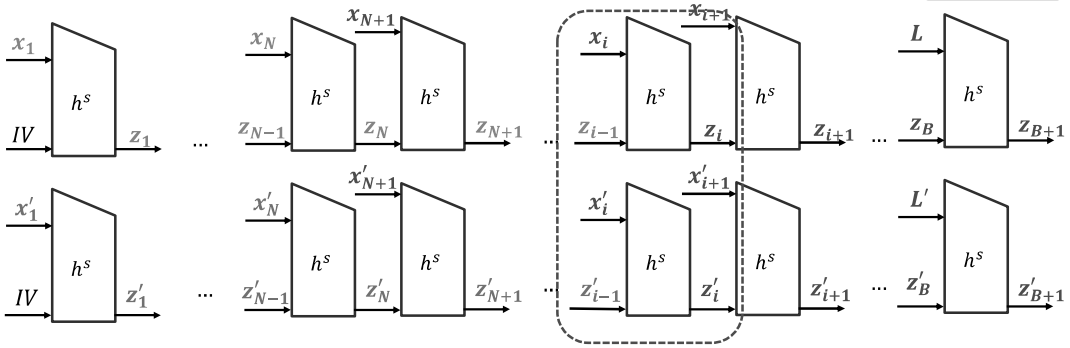
B $z_{N-1}||x_N$ 与 $z'_{N-1}||x'_N$

D $z_{i-1}||x_i$ 与 $z'_{i-1}||x'_i$

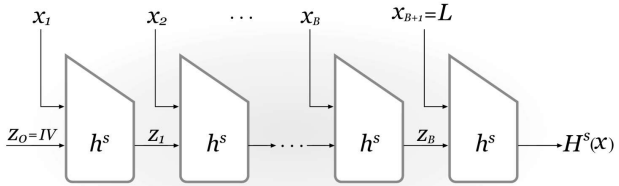
Let N ∈ {1, …, B} be the **largest** index such that $x_N \neq x'_N$

There must be $i \geq N$ such that $z_{i-1} \neq z'_{i-1}$ but $z_i = z'_i$

提交



The Merkle-Damgård Transform



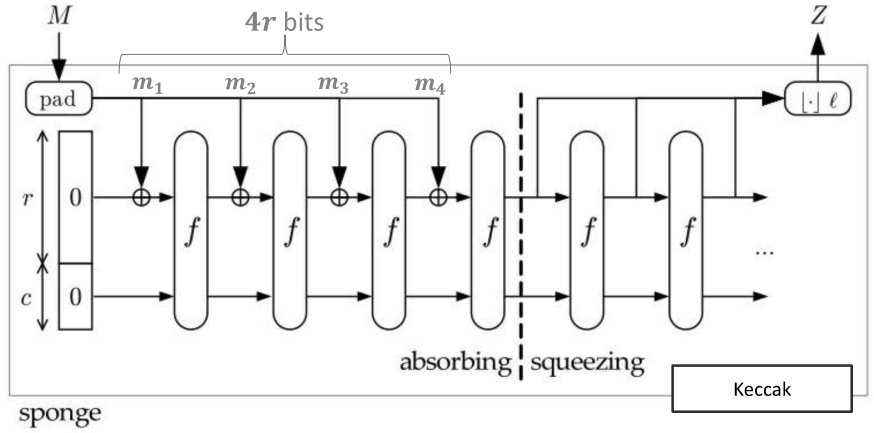
常见的哈希函数：SHA-1, **SHA-2 (SHA-256)**, MD5, **SM3**

SHA-3: Secure Hash Algorithm version 3

– NIST announced KECCAK as the winner of the SHA-3 Cryptographic Hash Algorithm Competition on October 2, 2012

Algorithm and variant	
MD5 (as reference)	
SHA-0	
SHA-1	
SHA-2	SHA-224
	SHA-256
	SHA-384
	SHA-512
	SHA-512/224
	SHA-512/256
SHA-3	SHA3-224
	SHA3-256
	SHA3-384
	SHA3-512
	SHAKE128
	SHAKE256

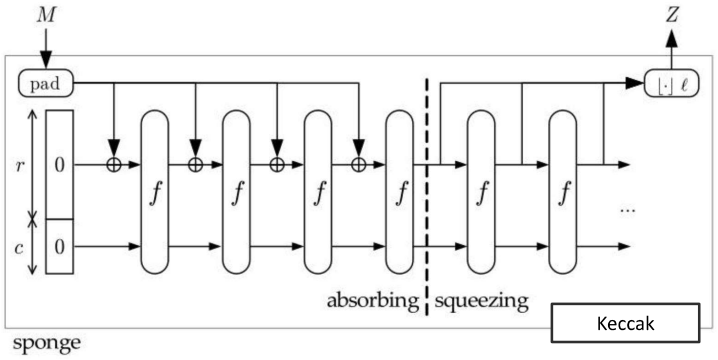
SHA-3: The Sponge Transform



$f: \{0,1\}^{r+c} \rightarrow \{0,1\}^{r+c}$ be a **permutation** function

(instead of a compression function)

SHA-3: The Sponge Transform



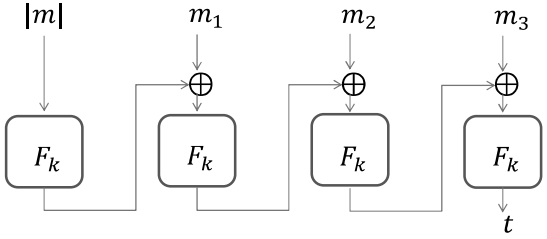
优点：1. 安全度**更高**（不仅局限于抗碰撞性）
2. 可输出**任意比特长度**
3. 抵御更强力的**侧信道攻击**

缺点：1. 效率低
2. 对f的安全性要求较高

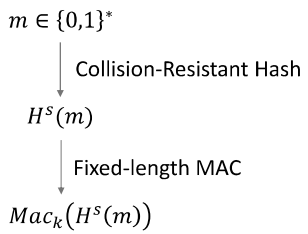
Apply hash functions on MAC

Domain Extension for MAC

- Improved CBC-MAC
  PRF → **Arbitrary** length MAC



- **Hash-then-MAC**
  Fixed-length MAC + CRH
  → **Arbitrary** length MAC



Hash-then-MAC Paradigm

CONSTRUCTION 5.5

Let $\Pi = (\mathsf{Mac}, \mathsf{Vrfy})$ be a MAC for messages of length $\ell(n)$, and let $\Pi_H = (\mathsf{Gen}_H, H)$ be a hash function with output length $\ell(n)$. Construct a MAC $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ for arbitrary-length messages as follows:

- Gen': on input 1^n, choose uniform $k \in \{0,1\}^n$ and run $\mathsf{Gen}_H(1^n)$ to obtain s; the key is $k' := \langle k, s \rangle$.
- Mac': on input a key $\langle k, s \rangle$ and a message $m \in \{0,1\}^*$, output $t \leftarrow \mathsf{Mac}_k(H^s(m))$.
- Vrfy': on input a key $\langle k, s \rangle$, a message $m \in \{0,1\}^*$, and a MAC tag t, output 1 if and only if $\mathsf{Vrfy}_k(H^s(m), t) \stackrel{?}{=} 1$.

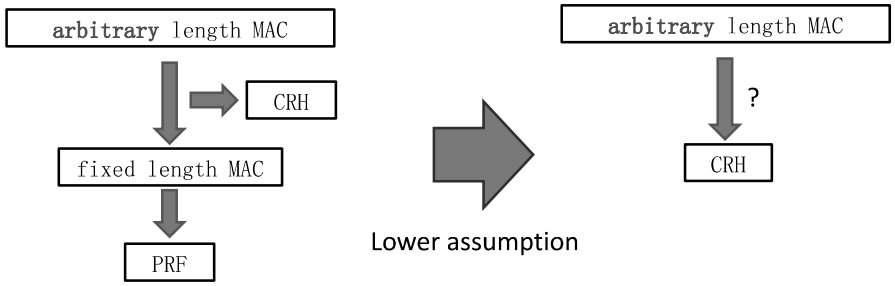
THEOREM 5.6 *If Π is a secure MAC for messages of length ℓ and Π_H is collision resistant, then Construction 5.5 is a secure MAC (for arbitrary-length messages).*

Proof Sketch of hash-then-MAC

- Let (m', t') be the forgery from the adversary A. Consider two cases:

1. If there exists a query m such that $H^s(m') = H^s(m)$, then (m, m') is a **collision** of H^s, then A can be used to break the collision resistance of H.
2. If there does not exist a query m such that $H^s(m') = H^s(m)$, then $\mathrm{Mac}\big(H^s(m')\big)$ is never computed in the queries. It breaks the security of $\Pi = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$.

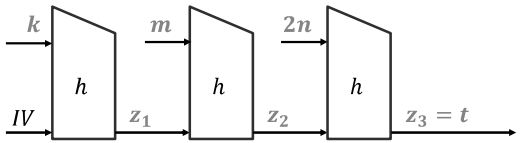
Construction of MAC



Construction of MAC

- 尝试：设H是抗碰撞哈希函数（如Merkle-Damgård结构）

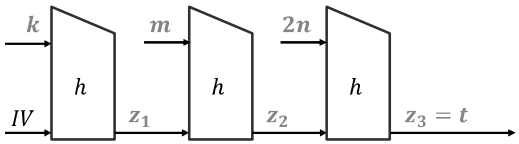
$Mac_k(m) \coloneqq H(k||m)$ 是安全的MAC吗？



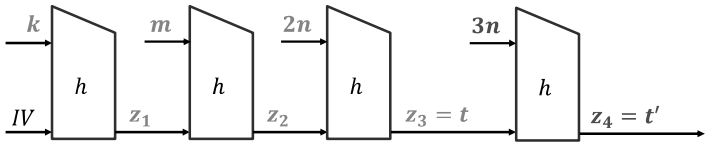
Construction of MAC

- 尝试：设H是抗碰撞哈希函数（如Merkle-Damgård结构）

$Mac_k(m) \coloneqq H(k||m)$ 是安全的MAC吗？



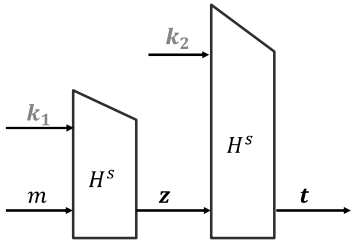
- 伪造：$Mac_k(m||2n) = H(k||m||2n) = h(t||\mathbf{3}n)$



NMAC: Secure MAC only based on CRH

- Let (Gen, H) be a collision-resistant hash function.

Given **two** random keys $k_1, k_2 \leftarrow \{0,1\}^n$ and hash key $s \leftarrow Gen(1^n)$, compute $Mac_k(m)$ as follows:

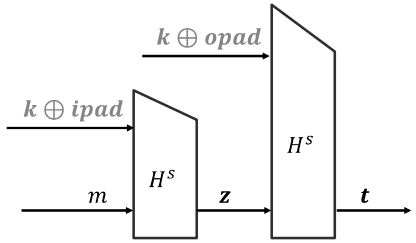


缺点：需要两个密钥

HMAC: Secure MAC only based on CRH

- Let (Gen, H) be a collision-resistant hash function.

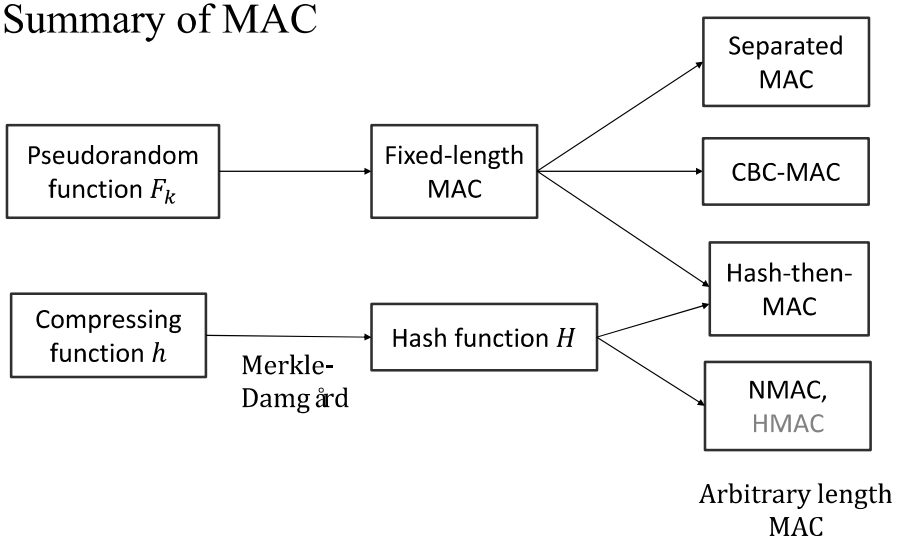
Given a random key $k \leftarrow \{0,1\}^n$ and a hash key $s \leftarrow Gen(1^n)$, compute $Mac_k(m)$ as follows:



- RFC2012标准MAC算法
- 国密算法HMAC-SM3
- Java密码库

(***ipad*** and ***opad*** are constant strings)

Summary of MAC



Homework P190

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(a) Prove that if at least one of (Gen_1, H_1) and (Gen_2, H_2) is collision resistant, then (Gen, H) is collision resistant.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. **（提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）此处H_1与H_2不必满足抗碰撞性。**

5.3 Let (Gen, H) be a collision-resistant hash function. Is $(\mathsf{Gen}, \hat{H})$ defined by $\hat{H}^s(x) \stackrel{\text{def}}{=} H^s(H^s(x))$ necessarily collision resistant?

Attacks on Hash & MAC

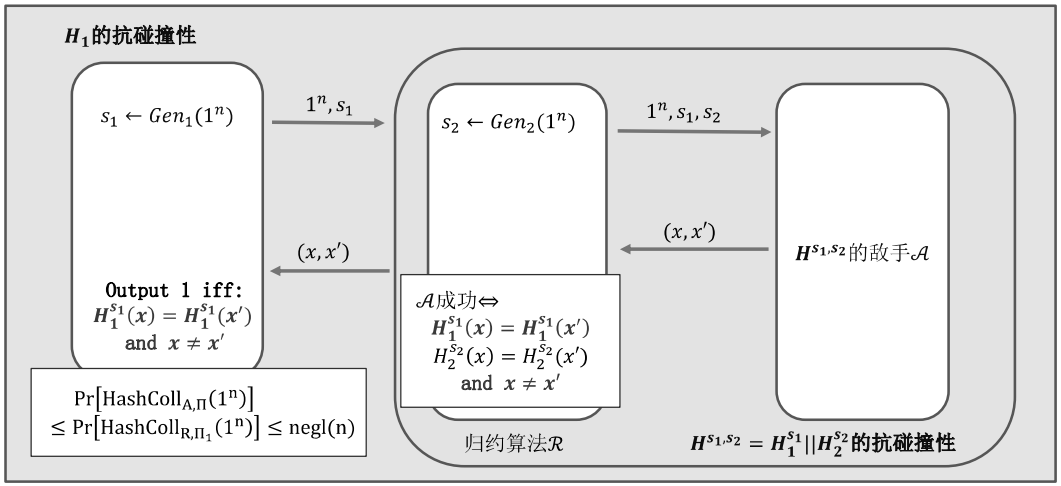
Homework

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(a) Prove that if at least one of (Gen_1, H_1) and (Gen_2, H_2) is collision resistant, then (Gen, H) is collision resistant.

假设H_1是满足抗碰撞性，
构造归约算法R，利用H的敌手A来攻击H_1的抗碰撞性

Homework



Homework

压缩性！

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. **（提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）**

设$H_2(x) = x$，此时$H(x) = H_1(x)||x$，敌手可以轻易通过$H(x)$得到x

而$H_2: \{0,1\}^l \rightarrow \{0,1\}^l$，不满足压缩性

压缩性！

Homework

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. （提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）

设$H_2: \{0,1\}^{l+1} \to \{0,1\}^l$ $H_2(b||x) := x$ for any $x \in \{0,1\}^l$ Or $H_2(x) := x \bmod 2^l$

此时H_2不抗原像，敌手可以通过找到H_2的原像来攻击H的抗原像性

压缩性！

Homework

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. （提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）

设$H_2: \{0,1\}^{l+1} \to \{0,1\}^l$ $H_2(b||x) := x$ for any $x \in \{0,1\}^l$ Or $H_2(x) := x \bmod 2^l$

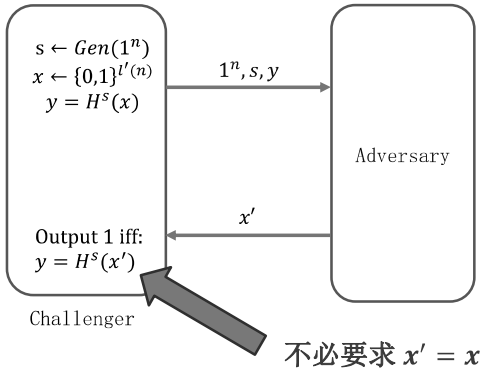
此时H_2不抗原像，敌手可以通过找到H_2的原像来攻击H的抗原像性

敌手真的可以通过$H_2(b||x)$得到原像 $b||x$ 吗？

敌手只能得到 x 而得不到 b!

Security Models

抗原像性 Preimage resistance



不必要求 $x' = x$

许多情况下，即便是无穷能力的敌手也无法找到 $x' = x$

压缩性！

Homework

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. （提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）

设$H_2: \{0,1\}^{l+1} \to \{0,1\}^l$ $H_2(b||x) := x$ （此处Gen_2无随机性，故省略s_2）

原像攻击：

1. 得到 s_1 与 $H(m) = y_1||y_2$，其中$y_1, y_2 \in \{0,1\}^l$
2. 对于$b \in \{0,1\}$, 令$m_b := b||y_2$
3. 分别计算$H_1^{s_1}(m_b)$

   若$y_1 = H_1^{s_1}(m_b)$，则输出m_b为$H(m)$的原像

如果$H_2: \{0,1\}^{2l} \to \{0,1\}^l$？

压缩性！

Homework

5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. （提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）

设$H_2: \{0,1\}^{2l} \to \{0,1\}^l$ $H_2(a||x) := x$ $a, x \in \{0,1\}^l$

原像攻击：
1. 得到 s_1 与 $H(m) = y_1||y_2$，其中$y_1, y_2 \in \{0,1\}^l$
2. 对于$a' \in \{0,1\}^l$, 令$m_{a'} := a'||y_2$
3. 分别计算$H_1^{s_1}(m_{a'})$ ⟸ 指数级时间！

   若$y_1 = H_1^{s_1}(m_{a'})$，则输出$m_{a'}$为$H(m)$的原像

压缩性！

Homework

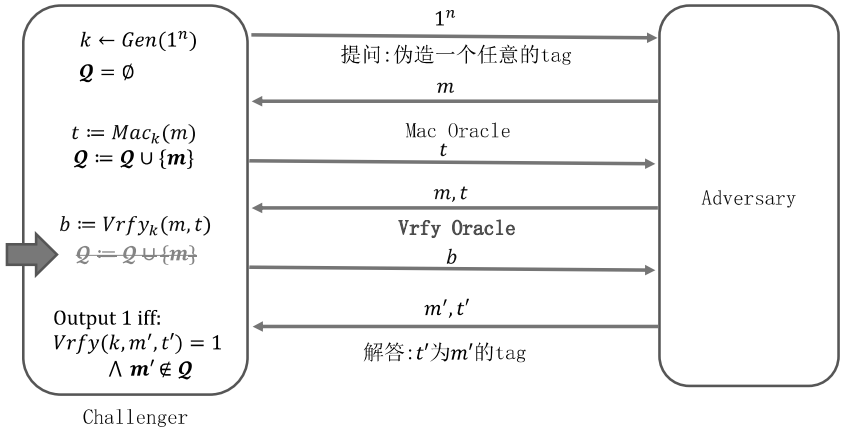
5.2 Let (Gen_1, H_1) and (Gen_2, H_2) be two hash functions. Define (Gen, H) so that Gen runs Gen_1 and Gen_2 to obtain keys s_1 and s_2, respectively. Then define $H^{s_1,s_2}(x) = H_1^{s_1}(x) \| H_2^{s_2}(x)$.

(b) Determine whether an analogous claim holds for ~~second preimage resistance and~~ preimage resistance, respectively. Prove your answer in each case. （提示：设H_1满足抗原像性，举反例H_2，使所得的H不满足抗原像性。）

思考：如果 (Gen_1, H_1) 和 (Gen_2, H_2) **都是**抗原像性的，

那么 (Gen, H) 一定抗原像性吗？

Homework: Security Model with Vrfy Oracle



Homework: Security Model with Vrfy Oracle

- 思路：
- 如果$m' \notin Q'$但$m' \in Q$，即敌手的伪造输入过Vrfy预言机而未输入过Mac预言机，则归约证明失败
- 事实上敌手对Vrfy预言机的问询并没有“实质作用”：

1. 如果敌手向Vrfy问到了一个合法的新tag（**新tag：不是从之前对Mac的问询中得来的**），则敌手这步操作本质上就是一个伪造，概率可忽略。
2. 如果敌手从未向Vrfy问过合法的新tag，敌手从Vrfy得到的信息就全部是0（旧tag除外），而问旧tag对敌手也没有什么帮助，所以敌手从Vrfy得不到任何**额外信息**。⟸ 得不到额外信息就意味着归约算法可能可以不借助外力而回答Vrfy！

◆ **关键事件：敌手是否问过合法的新tag！**

Homework: Security Model with Vrfy Oracle

- 定义事件：
- ValidQuery：敌手在攻击过程中问询过某个合法的(m,t)，且$m \notin Q$
- ValidQuery_i，该事件代表着第i次问询中**首次**触发 ValidQuery
  其中$i \in \{1,\dots,q(n)\}$，$q(n)$为敌手问询Vrfy预言机的次数上限。

- 原实验可分解为：

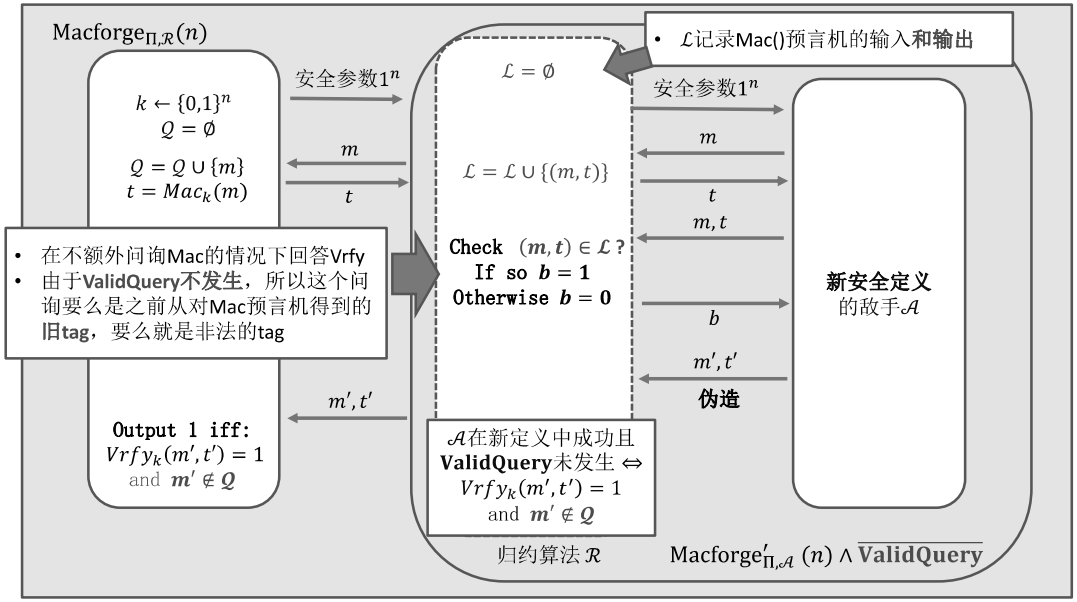
$$\Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n)=1\right] = \Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n)=1 \wedge \text{ValidQuery}\right] + \Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n)=1 \wedge \overline{\text{ValidQuery}}\right]$$

$$\leq \sum_{i=1}^{q(n)} \Pr[\text{ValidQuery}_i] + \Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n)=1 \wedge \overline{\text{ValidQuery}}\right]$$

Homework: Security Model with Vrfy Oracle

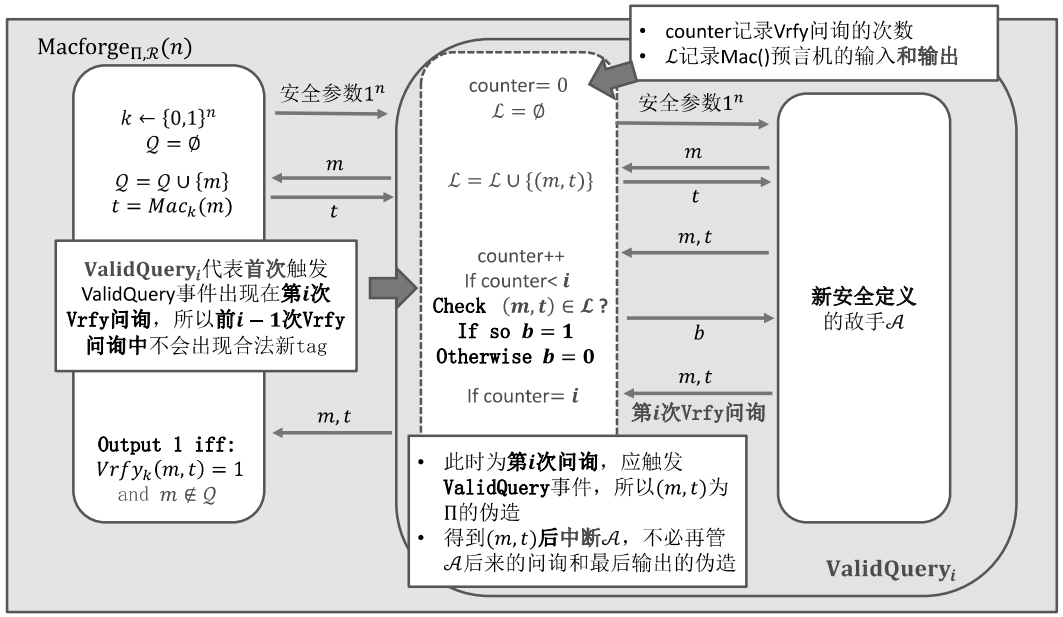
- Step 1：$\Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n)=1 \wedge \overline{\text{ValidQuery}}\right]$ is negligible
- 对任意P.P.T. 敌手$\mathcal{A}$，构造 $\mathcal{R}$ 攻击原 Macforge：

  关键思路：在归约算法从敌手收到对Vrfy的问询(m,t)时，想办法通过其他形式来回答，而不再借助问询Mac().



Homework: Security Model with Vrfy Oracle

- Step 2： $\Pr[\text{ValidQuery}_i]$ is negligible
- 对任意P.P.T. 敌手$\mathcal{A}$，构造$\mathcal{R}_i$攻击原Macforge：

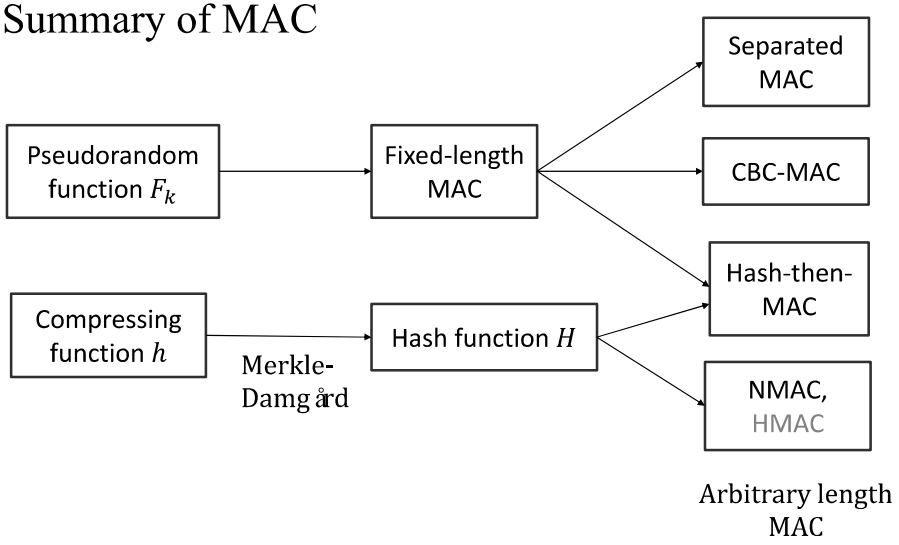


Homework: Security Model with Vrfy Oracle

- 对任意P.P.T. 敌手$\mathcal{A}$，都存在可忽略函数negl，满足：
  - $\Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n) = 1 \wedge \overline{\text{ValidQuery}}\right] \leq \Pr\left[\text{Macforge}_{\Pi,\mathcal{R}}(n) = 1\right] \leq negl(n)$
  - $\Pr[\text{ValidQuery}_i] \leq \Pr\left[\text{Macforge}_{\Pi,\mathcal{R}_i}(n) = 1\right] \leq negl(n)$
- $\Pr\left[\text{Macforge}'_{\Pi,\mathcal{A}}(n) = 1\right] \leq (q(n)+1) \cdot negl(n)$ ，其中$q(n)$为$\mathcal{A}$问询Vrfy预言机的次数。其为可忽略函数

安全损失（Security Loss）

Summary of MAC



Attacks on Hash and MAC

Attacks on MAC

- 既然我们已经给出了安全性证明，为什么还要考虑对MAC的攻击？

渐进性定义 Asymptotic Approach

- 基于安全参数进行定义，用复杂度（Complexity）而非确切数值刻画安全性。
- 有效的时间：多项式时间（Polynomial-time adversary）
- 极低的概率：可忽略概率（Negligible probability）

Attacks on MAC

- 理论安全性与具体安全性

2025年2月国家商用密码标准研究院《密码算法提交要求》

3.3.3 安全性声明与分析

算法提交者应给出算法的安全性声明，并从以下方面进行分析：

（1）理论安全性：应给出算法的安全模型，并证明算法的安全性。鼓励给出量子计算模型下的安全性证明。 ⬅ 渐进性安全证明

（2）具体安全性：针对算法声明的各种安全强度，应分别给出每个参数集抵抗已知经典计算攻击和量子计算攻击的时间复杂度。

⬆ 确定性安全强度衡量

Birthday Attack on CRH

- 生日问题 Gideon Yuval 1979
  - 问题：假设有q (q<365)个人，他们的生日在一年（N=365天）中平均分布，为保证这q个人中，至少有两人生日相同的概率大于1/2，q应取多大？
  - 答案：q=23 $\approx \sqrt{N}$

LEMMA A.15 *Fix a positive integer N, and say q elements $y_1, \ldots, y_q$ are chosen uniformly and independently at random from a set of size N. Then the probability that there exist distinct i, j with $y_i = y_j$ is at most $\frac{q^2}{2N}$. That is,*

$$\frac{1}{2} \leq \text{coll}(q, N) \leq \frac{q^2}{2N}. \quad \Rightarrow \quad q \geq \sqrt{N} = 19.1$$

仅代表生日攻击所需q的下界，不代表q的确切取值

Birthday Attack on CRH

- 生日问题 Gideon Yuval 1979
  - 问题：假设有q (q<365)个人，他们的生日在一年（N=365天）中平均分布，为保证这q个人中，至少有两人生日相同的概率大于1/2，q应取多大？

设 E_q 为q个人中（q≥2）每个人生日**都不相同**的事件

$$\Pr[E_2] = 1 - \frac{1}{N} = \frac{N-1}{N}$$

$$\Pr[E_3] = \Pr[E_2] \cdot \Pr[y_3 \notin \{y_1, y_2\} | E_2] = \frac{N-1}{N} \cdot \frac{N-2}{N}$$

$$\Pr[E_q] = \Pr[E_{q-1}] \cdot \Pr[y_q \notin \{y_1, y_2\} | E_{q-1}] = \frac{N-1}{N} \cdot \frac{N-2}{N} \cdot \ldots \cdot \frac{N-q+1}{N}$$

$$= \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$$

Birthday Attack on CRH

- 生日问题 Gideon Yuval 1979
  - 问题：假设有q (q<365)个人，他们的生日在一年（N=365天）中平均分布，为保证这q个人中，至少有两人生日相同的概率大于**1/2**，q应取多大？

    设 E_q为q个人中（q≥2）每个人生日**都不相同**的事件

$$\Pr[E_q] = \prod_{i=1}^{q-1}\left(1-\frac{i}{n}\right)$$

$$\ln \Pr[E_q] = \sum_{i=1}^{q-1} \ln\left(1-\frac{i}{n}\right) \approx \sum_{i=1}^{q-1}\left(-\frac{i}{n}\right) = -\frac{1}{n}\sum_{i=1}^{q-1} i = -\frac{q(q-1)}{2n} < -\ln 2$$

$$q \approx 1.17\sqrt{N}$$

Birthday Attack on CRH

- Assume CRH: $\{0,1\}^* \to \{0,1\}^l$
- A birthday attack finds a collision with high probability in

  $O(q) = O(\sqrt{2^l}) = O(2^{l/2})$time

  and $O(l\, 2^{l/2})$ space

x	H(x)
000000000	01001001
000000001	00111010
…	…
011001010	00111010

$\approx 2^{l/2}$

- Finding a collision (x, x') immediately implies a chosen message attack on **Hash-then-MAC** (or)paradigm.
- $t = MAC_k(H(m))$

  Query x to MAC oracle and forge for x'

≠ finding (second-)preimage

Finding Meaningful Collisions

- Find a pair of collision (x, x') with **meaningful** messages.

It is *hard/difficult/challenging/impossible* to *imagine/believe* that we will *find/locate/hire* another *employee/person* having similar *abilities/skills/character* as Alice. She has done a *great/super* job.

Thus, the sentence can be written in $4 \cdot 2 \cdot 3 \cdot 2 \cdot 3 \cdot 2 = 288$ different ways.

x	H(x)
Good message 1	01001001
Good message 2	00111010
…	…
Good message $2^{l/2}$	00111010

x'	H(x')
Bad message 1	01011100
Bad message 2	11110011
…	…
Bad message $2^{l/2}$	10010100

Find a collision from the both sides

Birthday Attack on CRH

- Require a large amount of memory
  - it is far more feasible to run in 2^64 time than it is to obtain a disk of size 2^64
    - 2^60 Bytes: 1 billion gigabytes,

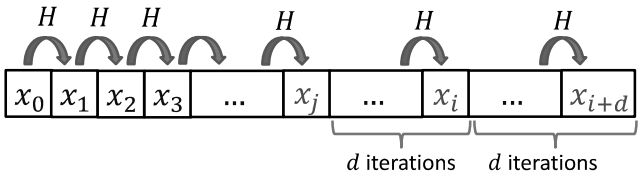
      1 million Terabytes
    - 2^60 CPU instructions: 2 year

x	H(x)
000000000	01001001
000000001	00111010
…	…
011001010	00111010

$\approx 2^{l/2}$

Improved Birthday Attack on CRH

- Randomly choose $x_0 \in \{0,1\}^{l+1}$ and let $x_i = H(x_{i-1})$



$$H(x_0) = x_1$$
$$H(x_1) = x_2$$
$$H(x_2) = x_3$$
$$H(x_3) = x_4$$
$$\vdots$$
$$H(x_{i-1}) = x_i$$

Let $x_j = x_i$ be the first pair of identical string $(1 \le j \le i)$

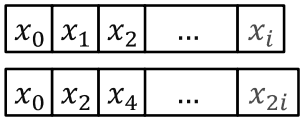
$x_j = x_i \Leftrightarrow H(x_{j-1}) = H(x_{i-1}) \Leftrightarrow (x_{j-1}, x_{i-1})$ is a collision

We need to **record** all $x_0 \sim x_i$ to find i, j !

Improved Birthday Attack on CRH

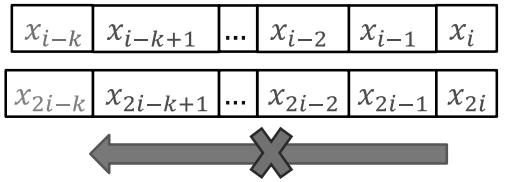
- Randomly choose $x_0 \in \{0,1\}^{l+1}$ and let $x'_0 = x_0$
- For $i = 1,2,\ldots$, $x_i = H(x_{i-1})$, $x'_i = H(H(x'_{i-1})) = x_{2i}$

$$H(x_0) = x_1$$
$$H(x_1) = x_2$$
$$H(x_2) = x_3$$
$$H(x_3) = x_4$$
$$\vdots$$
$$H(x_{i-1}) = x_i$$



只需动态存储每一对x_i, x'_i

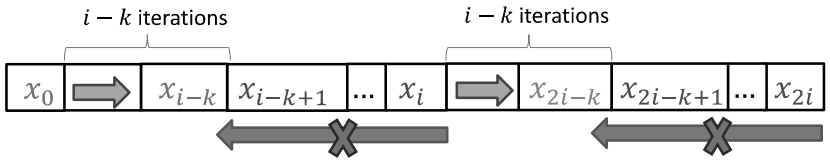
Suppose $x_i = x_{2i}$, then $H(x_{i-1}) = H(x_{2i-1})$. But it is possible that $x_{i-1} = x_{2i-1}$



Trace back k iterations
until $x_{i-k} \neq x_{2i-k}$

However, we still need to record **all** $x_0 \sim x_{2i}$ for tracing back!

Improved Birthday Attack on CRH



- Note that $x_0 \in \{0,1\}^{l+1}$, we have $x_0 \neq x_i$ ($x_i \in \{0,1\}^l$).
- There must be $\boldsymbol{k \le i}$ such that $x_{i-k} \neq x_{2i-k}$.

➢ Our Collision-finding algorithm runs as follows:

1. Find i and (x_i, x_{2i}) such that $x_i = x_{2i}$,
2. Find j **(indeed *j=i-k*)** such that $x_{j+1} = x_{i+j+1}$
3. Output (x_j, x_{i+j})

Improved Birthday Attack on CRH

ALGORITHM 5.9
A small-space birthday attack

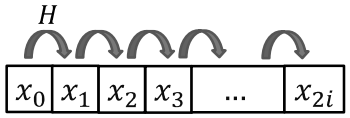
Input: A hash function $H : \{0,1\}^* \to \{0,1\}^\ell$
Output: Distinct x, x' with $H(x) = H(x')$

$x_0 \leftarrow \{0,1\}^{\ell+1}$
$x' := x := x_0$ ≈ $3l$ bit
for $i = 1, 2, \ldots$ **do**:
  $x := H(x)$
  $x' := H(H(x'))$
  // now $x = H^{(i)}(x_0)$ and $x' = H^{(2i)}(x_0)$
  if $x = x'$ **break**
$x' := x$, $x := x_0$ ≈ $O(2^{l/2})$ loops
for $j = 1$ to i:
  if $H(x) = H(x')$ **return** x, x' and **halt**
  else $x := H(x)$, $x' := H(x')$
  // now $x = H^{(j)}(x_0)$ and $x' = H^{(i+j)}(x_0)$

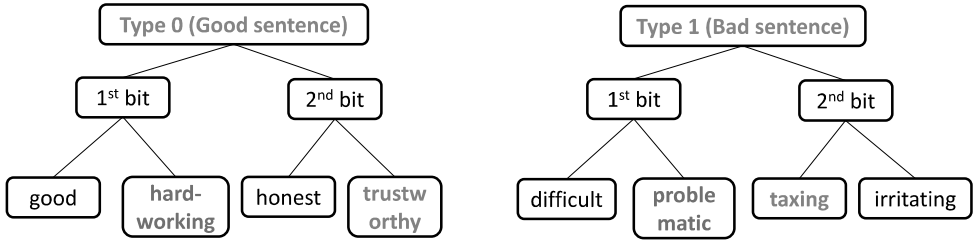
➢ Space Complexity: $\Theta(l)$

➢ Time Complexity: $\Theta(2^{l/2})$

Finding meaningful collision

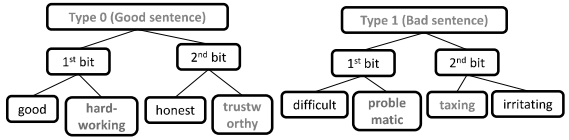


- All the x_i's are probably meaningless.
- Solution: Define a "**message function**"!



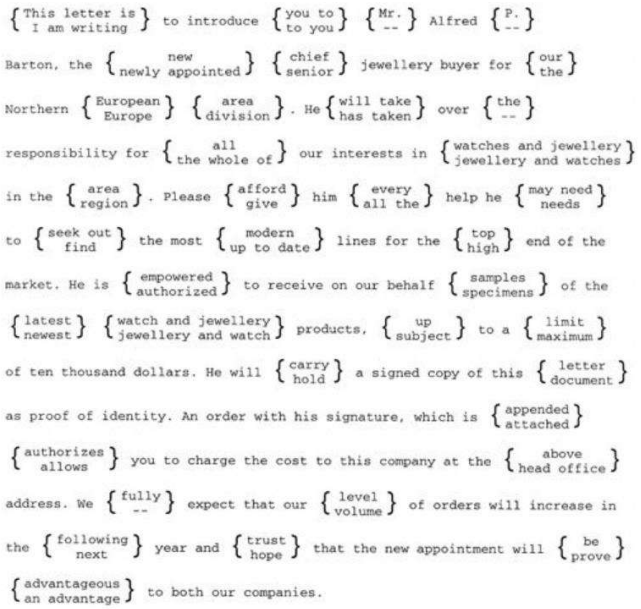
Example: $f(110)$ =Alice is a hard-working and trustworthy student.
$f(101)$ =Alice is a problematic and taxing student.

Finding meaningful collision



- Find a collision for $\boldsymbol{g(x) := H(f(x))}$
- E.g., $(110,101)$ is a collision of $g(\mathrm{x})$.
  Then, "Alice is a hard−working and trustworthy student." and
  "Alice is a problematic and taxing student." is a collision of H.

- With probability 1/2, x and x' are from two different types.
  Then, $f(x)$ and $f(x')$ are meaningful collisions of H.



{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}
Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}
Northern {European / Europe} {area / division}. He {will take / has taken} over {the / --}
responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}
in the {area / region}. Please {afford / give} him {every / all the} help he {may need / needs}
to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the
market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the
{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}
of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}
as proof of identity. An order with his signature, which is {appended / attached}
{authorizes / allows} you to charge the cost to this company at the {above / head office}
address. We {fully / --} expect that our {level / volume} of orders will increase in
the {following / next} year and {trust / hope} that the new appointment will {be / prove}
{advantageous / an advantage} to both our companies.

A letter with 2^{37} different types of expressions [DAVI89]

A Concrete Attack

- Eve准备两份合同，一份（F1）对Alice有利，另一份（F2）能够让Alice破产
- Eve对每份合同做细微修改，并分别计算新的Hash值（如在一行末尾打上一个或两个空格；添加逗号等，在不改变原文含义的基础上，轻易生成2^{32}个不同的合同）
- 在生成的两个Hash值集合中，Eve寻找Hash值相等的两份合同（F1'，F2'）（若Hash值的长度为**64比特**， 根据生日攻击，可以使用大约**2^{32}**次Hash计算便可以以大于1/2的概率找到这样一对碰撞）
- Eve将对Alice有利的合同（F1'）发送给Alice让她签署（协议假设Alice只需签署Hash值）
- 一旦Alice签署完成，Eve可以替换为合同（F2'），并向Bob证明，合同（F2'）的确是Alice签署的。

Other Attacks

- Sec 5.4.3 Time/Space Tradeoffs for Inverting Functions
  (Attacking Preimage Resistance.)

- Next Week: Applications of Hash and MAC

Applications of Hash & MAC

Review

- Birthday Attack on CRH & MAC: $\{0,1\}^* \rightarrow \{0,1\}^l$

- Improved Birthday Attack:
  Time Complexity: $\Theta(2^{l/2})$. Space Complexity: $\Theta(l)$
- Meaningful Collisions
- Concrete Attacks on MAC

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Rounds	Operations	Security against collision attacks (bits)	Security against length extension attacks (bits)	Performance on Skylake (median cpb)[1] Long messages	Performance on Skylake (median cpb)[1] 8 bytes	First published
MD5 (as reference)		128	128 (4 × 32)	512	4 (16 operations in each round)	And, Xor, Or, Rot, Add (mod 2^{32})	≤ 18 (collisions found)[2]	0	4.99	55.00	1992
SHA-0		160	160 (5 × 32)	512	80	And, Xor, Or, Rot, Add (mod 2^{32})	< 34 (collisions found)	0	≈ SHA-1	≈ SHA-1	1993
SHA-1							< 63 (collisions found)[3]		3.47	52.00	1995
SHA-2	*SHA-224*	224	256 (8 × 32)	512	64	And, Xor, Or, Rot, Shr, Add (mod 2^{32})	112	32	7.62	84.50	2004
	SHA-256	256					128	0	7.63	85.25	2001
	SHA-384	384	512 (8 × 64)	1024	80	And, Xor, Or, Rot, Shr, Add (mod 2^{64})	192	128	5.12	135.75	2001
	SHA-512	512					256	0[4]	5.06	135.50	2001
	SHA-512/224	224					112	288	≈ SHA-384	≈ SHA-384	2012
	SHA-512/256	256					128	256			
SHA-3	*SHA3-224*	224	1600 (5 × 5 × 64)	1152	24[5]	And, Xor, Rot, Not	112	448	8.12	154.25	2015
	SHA3-256	256		1088			128	512	8.59	155.50	
	SHA3-384	384		832			192	768	11.06	164.00	
	SHA3-512	512		576			256	1024	15.88	164.00	
	SHAKE128	*d* (arbitrary)		1344			min(*d*/2, 128)	256	7.08	155.25	
	SHAKE256	*d* (arbitrary)		1088			min(*d*/2, 256)	512	8.59	155.50	

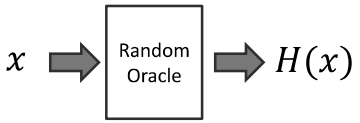
n-bit Security:
最优攻击需要
$\approx 2^n$ 次哈希运算

Random Oracle Model

Random Oracle Model (随机预言机模型, ROM)

- 最理想化的哈希函数模型：将哈希函数理想化为**真随机函数**
- 真随机函数$f(\cdot)$：
  - 对于任意未被计算过的x，$f(x)$的分布是**独立且均匀随机的**；
  - 和理想的哈希函数性质接近：
    要获取$H(x)$的相关信息，**只能通过计算H来获得**
    （而不能通过其他手段，如计算$H(x')$）

x	H(x)
00000000	01001000
00000001	11010101
00000010	11001010
.....	.....
11111111	01011100



（黑盒子）

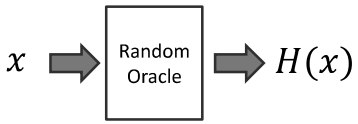
Random Oracle Model (随机预言机模型, ROM)

- 最理想化的哈希函数模型：将哈希函数理想化为**真随机函数**

不是由挑战者和敌手组成的**安全模型**！
是一种针对**哈希函数**的假设
（其他安全模型基础上的“额外规则”）

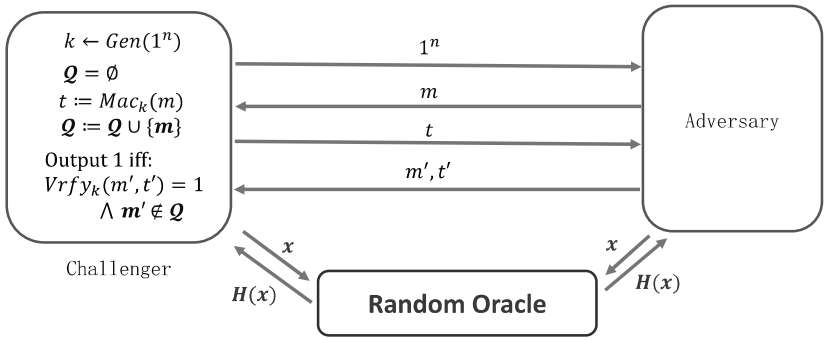
- 随机预言机模型的**规则（理想化模型）**：
- 假设存在一个**公共的**预言机：RO
- 当敌手/挑战者计算任意$H(x)$的值时，都**必须**向RO问询x
- 对于未被访问过的x， 在H值域上均匀随机取y
- 定义$y \coloneqq H(x)$

不能私下通过运行
H的代码来计算



Random Oracle Model (随机预言机模型, ROM)

- 举例：假设MAC中使用了哈希函数H（如Hash-then-MAC）**且H被模型化为RO**
- **随机预言机模型下的MAC存在性不可伪造模型 (EU-CMA in ROM)：**



Random Oracle Model (随机预言机模型, ROM)

- 若在其中哈希函数被模型化为随机预言机的条件下，完成了算法的安全性证明，则称该算法**在随机预言机模型下安全**。

If the security is proven when the hash function is modeled as a random oracle, then we say it is secure **in the random oracle models (ROM).**

- 否则（如仅要求哈希函数满足抗碰撞性），则称该算法**在标准模型下安全**。

Otherwise, we say it is secure **in the standard models.**

Random Oracle Model (随机预言机模型, ROM)

- **最理想化**的哈希函数模型：
- 在随机预言机模型下，**哈希函数一定满足抗碰撞性和抗原像性**。
- 证明：若$H:\{0,1\}^{l(n)} \to \{0,1\}^n$是随机预言机，则任意多项式敌手只能以可忽略的概率破坏抗原像性。

Random Oracle Model (随机预言机模型, ROM)

- 证明：若$H:\{0,1\}^{l(n)} \to \{0,1\}^n$被模型化为随机预言机，则任意多项式敌手只能以可忽略的概率破坏抗原像性。（不再反证）
- 挑战者：任取$x \leftarrow \{0,1\}^{l(n)}$，访问RO，得到$y = H(x)$，将$y$发送给敌手。
- 设敌手攻击过程中向RO访问过$x_1, \dots, x_q$，最终输出x'，分以下情况讨论：

1. 若$x' \notin \{x_1, \dots, x_q\}$，**敌手不知道$H(x')$的任何信息**，其**恰好**满足$y = H(x')$的概率为$1/2^n$。
2. 若$x' \in \{x_1, \dots, x_q\}$，设$y_1, \dots, y_q$分别为从RO所得到的回答，对于每一个$y_i$，其与$y$相等的概率**皆独立且等于$1/2^n$**，故$y = H(x')$的概率至多为$q/2^n$。

- 敌手成功攻击RO抗原像性的概率至多为 $(q+1)/2^n$。

Random Oracle Model (随机预言机模型, ROM)

- 优点：

1. **一句话**概括所有哈希函数的所需的性质，更简练。
2. 许多可证明安全**必须**要求随机预言机模型。

抗（第二）原像
抗碰撞
坍塌性等.....
VS
ROM

- 缺点：

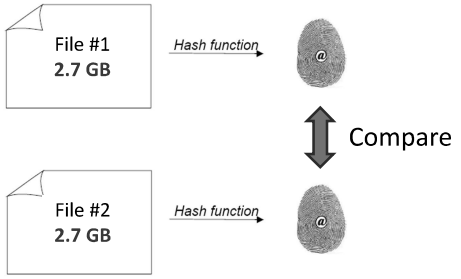
1. 对哈希函数的**安全性假设要求较高**，可信度较低。
   （如Merkle-Damgard结构**某些情况下**不能看作RO： Exercise 5.10）
2. 该模型的运行规则与**实际应用**场景有出入。
3. 某些**“极特殊”**的反例下，ROM下拥有可证明安全的算法在实际中并不安全。

- 总结：**瑕不掩瑜！**

Applications of Hash & MAC

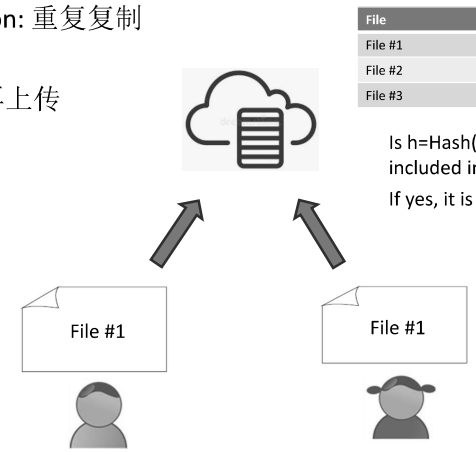
Fingerprinting and Deduplication

- Virus Fingerprinting: 校验指纹



Fingerprinting and Deduplication

- Deduplication: 重复复制
- 先校验，再上传



File	Hash
File #1	...
File #2	...
File #3	...

Is h=Hash(File #1) already included in the hash list?

If yes, it is unnecessary to upload it again!

Password Hashing

- 本地校验口令
- Password.ini:

  If password == 123456

  then login();

INSECURE！！

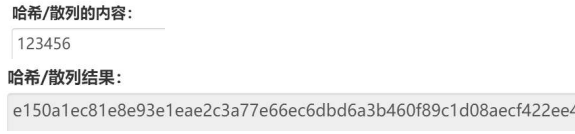
敌手可通过直接读取配置文件来窃取口令

- Password.ini:

  If SHA256(password) == e150a1...

  then login();

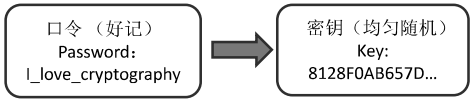
即便读取文件也无法获取口令信息



注：更安全的做法是使用SHA256(s||password)，一定程度上预防常用密码的查表攻击

Key Derivation （密钥派生）

- 加密和MAC密钥要求**均匀随机的**密钥：$k \leftarrow \{0,1\}^n$
- $\boldsymbol{key = Hash(PASSWORD)}$
- 若口令的**最小信息熵**大于 n （任意口令都**至多**以 2^{-n} 的几率被选中），
  且**Hash是RO模型**，则所得的key为均匀随机分布

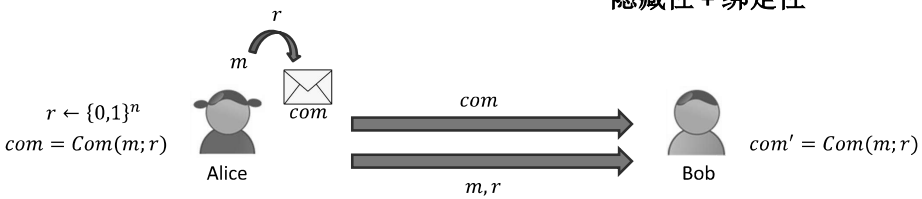


DEFINITION 5.12 *A probability distribution $\mathcal{X}$ has m* bits of min-entropy *if for every fixed value x it holds that $\Pr_{X \leftarrow \mathcal{X}}[X = x] \leq 2^{-m}$. That is, even the most likely outcome occurs with probability at most 2^{-m}.*

Commitment Scheme

- **承诺**算法（基本语法与安全性的定义不唯一）：
- Alice向Bob承诺某项陈述m (statement)， 但不暴露陈述的具体内容。
- Alice可以在**承诺后**向Bob提供陈述内容和证据，Bob验证其是否与承诺一致。

隐藏性 + 绑定性

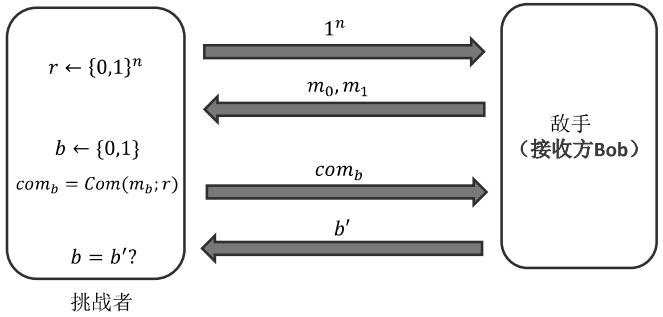


m = "My ID number is 370000200401011234"

Commitment Scheme

- **隐藏性（Hiding）**：承诺 com 不暴露陈述的内容 m （不可区分）

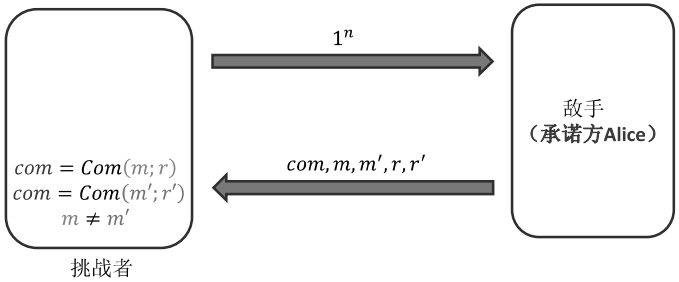
The commitment hiding experiment $\text{Hiding}_{\mathcal{A},\text{Com}}(n)$**:**



Commitment Scheme

- **绑定性（Binding）**：承诺者无法对已给的承诺“反悔”
  即无法对一个承诺打开两种不同的陈述

The commitment binding experiment $\text{Binding}_{\mathcal{A},\text{Com}}(n)$**:**



Commitment Scheme

DEFINITION 5.13 *A commitment scheme* Com *is* secure *if for all* PPT *adversaries* $\mathcal{A}$ *there is a negligible function* negl *such that*

$$\Pr\left[\mathsf{Hiding}_{\mathcal{A},\mathsf{Com}}(n)=1\right] \leq \frac{1}{2}+\mathsf{negl}(n)$$
and
$$\Pr\left[\mathsf{Binding}_{\mathcal{A},\mathsf{Com}}(n)=1\right] \leq \mathsf{negl}(n).$$

- Define: $\boldsymbol{Com(m;r) = H(m||r)}$
- Break Hiding ⇔ Inverse m from $H(m||r)$ ⇐ RO模型
- Break Binding ⇔ Find $m \neq m'$ and r, r' such that $H(m||r) = H(m'||r')$ ⇐ 抗碰撞性

Application of Hash Functions

- Constructing MAC for Message Integrity
- Fingerprinting and Deduplication
- Password Checking
- Key Derivation
- Constructing Commitment Schemes
- Merkle tree…
- Constructing **A LOT OF SCHEMES…**
  (including public-key schemes)

Application of MAC: Authenticated Encryption

- **加密：消息隐私性（不可区分）**
- **签名/MAC：消息完整性（不可伪造）**

- **加密 + MAC = 可认证加密（AE）**

Authenticated Encryption

- Let $\Pi = (Gen, Enc, Dec)$ be an authenticated encryption such that:
- $Gen(1^n)$ outputs a random key k.
- $Enc_k(m)$ outputs a ciphertext c for plaintext m.
- $Dec_k(c)$ outputs a plaintext m for ciphertext c **or an error symbol** $\perp$.

- For any k and m, it holds that $Dec_k\big(Enc_k(m)\big) = m \neq \perp$
- For some **invalid** ciphertexts c' generated though **abnormal means**, it may have $Dec_k(c') = \perp$

Authenticated Encryption

The unforgeable encryption experiment $\mathsf{Enc\text{-}Forge}_{\mathcal{A},\Pi}(n)$**:**

1. *Run* $\mathsf{Gen}(1^n)$ *to obtain a key* k.
2. *The adversary* $\mathcal{A}$ *is given input* 1^n *and access to an encryption oracle* $\mathsf{Enc}_k(\cdot)$. *The adversary outputs a ciphertext* c.
3. *Let* $m := \mathsf{Dec}_k(c)$, *and let* $\mathcal{Q}$ *denote the set of all queries that* $\mathcal{A}$ *asked its encryption oracle. The output of the experiment is* 1 *if and only if (1)* $m \neq \perp$ *and (2)* $m \notin \mathcal{Q}$.

- 敌手的目标是自行生成一个合法（且未被询问过的）的密文

Authenticated Encryption

The unforgeable encryption experiment $\mathsf{Enc\text{-}Forge}_{\mathcal{A},\Pi}(n)$**:**

1. *Run* $\mathsf{Gen}(1^n)$ *to obtain a key* k.
2. *The adversary* $\mathcal{A}$ *is given input* 1^n *and access to an encryption oracle* $\mathsf{Enc}_k(\cdot)$. *The adversary outputs a ciphertext* c.
3. *Let* $m := \mathsf{Dec}_k(c)$, *and let* $\mathcal{Q}$ *denote the set of all queries that* $\mathcal{A}$ *asked its encryption oracle. The output of the experiment is* 1 *if and only if (1)* $m \neq \perp$ *and (2)* $m \notin \mathcal{Q}$.

DEFINITION 4.16 *A private-key encryption scheme* Π *is* unforgeable *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$, *there is a negligible function* negl *such that:*

$$\Pr[\mathsf{Enc\text{-}Forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Authenticated Encryption

- If an Authenticated Encryption is secure under chosen plaintext attack and unforgeable, then it is secure under chosen ciphertext attack.
- 抗选择明文攻击 + 不可伪造 = 抗选择密文攻击

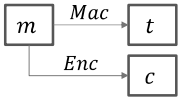
- 敌手能获取什么信息？
- 唯密文攻击
  敌手能获取某个（些）密文
- 已知密文攻击
  敌手额外知晓某些明密文对
- 选择明文攻击
  敌手额外拥有加密机，可以加密任意明文（目标明文除外）
- 选择密文攻击
  敌手额外拥有解密机，可以解密任意密文（目标密文除外）

Upgrade!

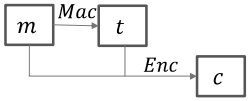
Authenticated Encryption

- Let Π_E be a CPA-secure encryption scheme and Π_M be a secure MAC. There are **3 approaches**:

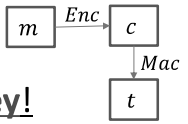
1. Encrypt-and-MAC:
   $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(m)$. Output (c, t)



2. MAC-then-Encrypt:
   $t \leftarrow Mac_{k_M}(m),\ c \leftarrow Enc_{k_E}(m||t)$. Output c



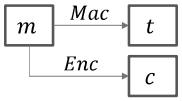
3. Encrypt-then-MAC:
   $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(c)$. Output (c, t)



Never do encryption and MAC with the same key!

Encrypt-and-MAC:

➢ $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(m)$. Output (c, t)



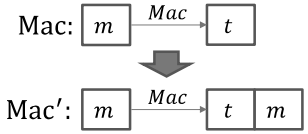
➢ t 可能会暴露 m 的信息！

➢ 反例：假设$(Gen, Mac, Vrfy)$是任意安全的MAC算法，那么以下算法$(Gen, Mac', Vrfy')$也是安全的：

$Mac'_k(m) := Mac_k(m)||m$

$Vrfy'_k(m, (m'||t))$: Output 1 iff $Vrfy_k(m, t) = 1$ and $m = m'$.

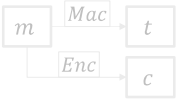
- 此时Encrypt-and-MAC是不安全的



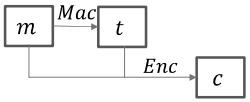
Authenticated Encryption

➢ Let Π_E be a CPA-secure encryption scheme and Π_M be a secure MAC. There are **3 approaches**:

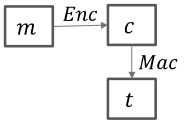
1. Encrypt-and-MAC:
   $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(m)$. Output (c, t)



2. MAC-then-Encrypt:
   $t \leftarrow Mac_{k_M}(m),\ c \leftarrow Enc_{k_E}(m||t)$. Output c



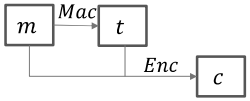
3. Encrypt-then-MAC:
   $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(c)$. Output (c, t)



学术界有一定争议，但一般选3

MAC-then-Encrypt:

➢ $t \leftarrow Mac_{k_M}(m),\ c \leftarrow Enc_{k_E}(m||t)$. Output c



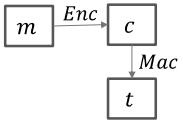
➢ 当Enc分组密码算法涉及到填充（Padding）时，在使用某些加密算法时，可能会导致安全隐患

➢ POODLE 攻击（Padding Oracle On Downgraded Legacy Encryption）
针对SSL3.0协议中使用的CBC加密 （2014年）

➢ 2015年，RFC 7568 标准弃用SSL 3.0。

Encrypt-then-MAC :

➢ $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(c)$. Output (c, t)



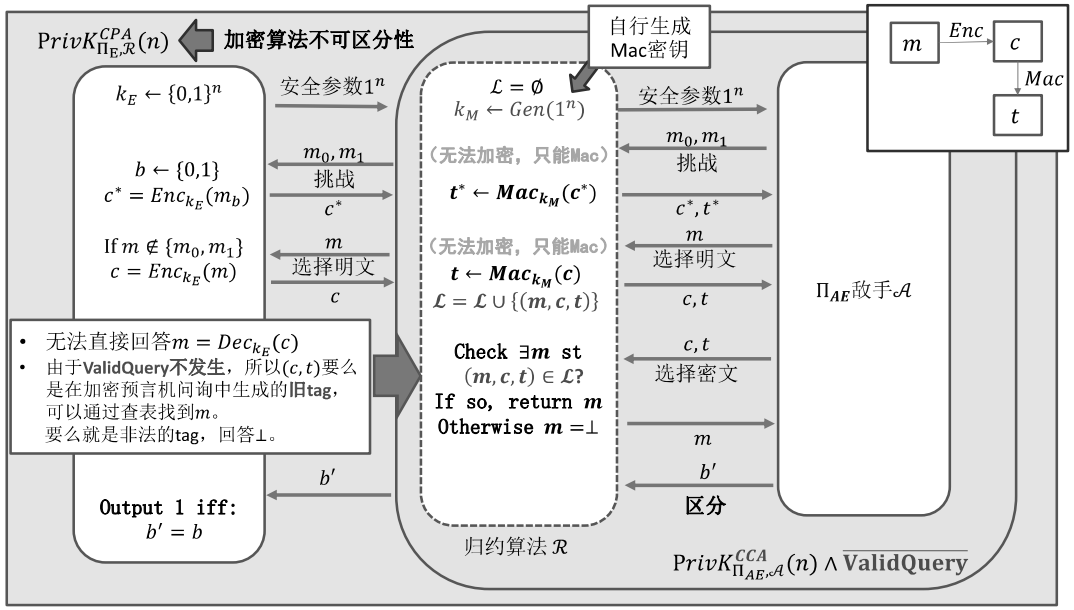
➢ **可证明安全！**

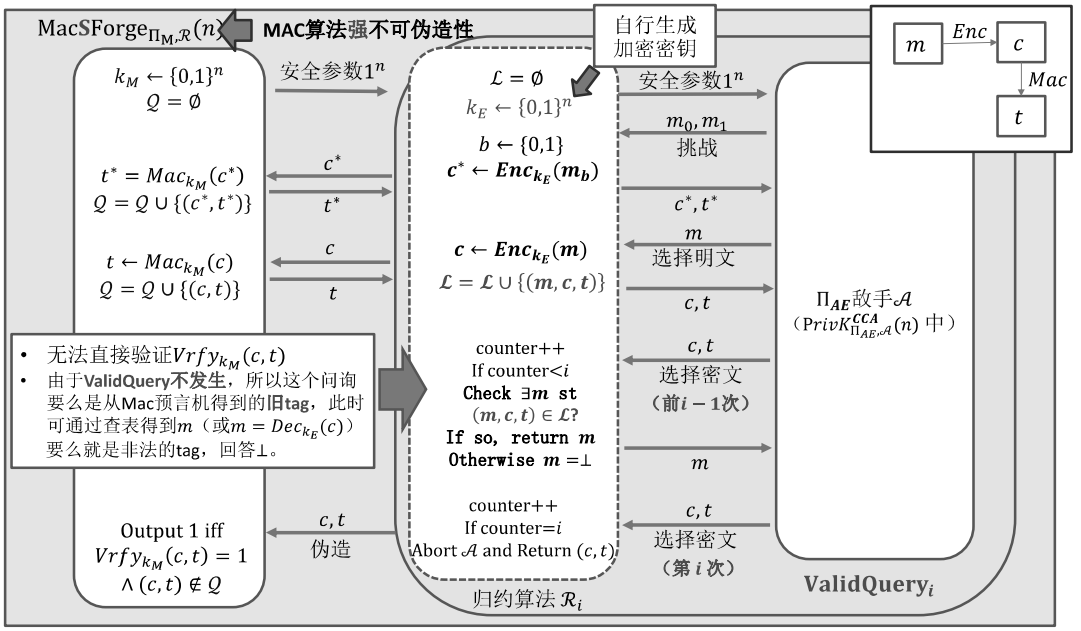
➢ **若**(Enc, Dec)是**选择明文攻击（CPA）**下不可区分的，**且**$(Gen, Mac, Vrfy)$是选择消息攻击下**强**不可伪造的，则以上算法是**选择密文攻击（CCA）**下不可区分的。（自选挑战）

➢ 强不可伪造性：不可伪造 $(m', t') \notin Q$

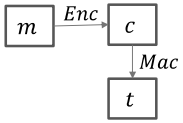
➢ 思路：类似第一次作业（第6讲），关键事件为敌手是否访问了**合法的新tag**

- **ValidQuery：敌手在某次对解密预言机的问询中，问过合法的新tag。**
- **ValidQuery$_i$：敌手在第i次对解密预言机的问询中，首次问了合法的新tag。**





Encrypt-then-MAC :



- $c \leftarrow Enc_{k_E}(m),\ t \leftarrow Mac_{k_M}(c)$. Output (c,t)

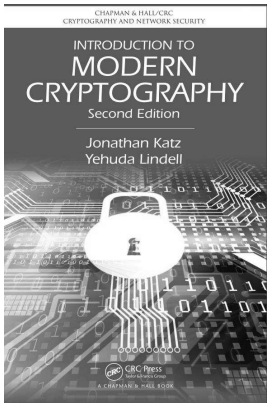
对任意P.P.T. 敌手$\mathcal{A}$，都存在可忽略函数negl，满足：

- $\Pr\left[PrivK^{\boldsymbol{CCA}}_{\Pi_{AE},\mathcal{A}}(n) \wedge \overline{\textbf{ValidQuery}}\right] \le \Pr\left[PrivK^{CPA}_{\Pi_E,\mathcal{R}}(n) = 1\right] \le \frac{1}{2} + negl(n)$
- $\Pr\left[PrivK^{\boldsymbol{CCA}}_{\Pi_{AE},\mathcal{A}}(n) \wedge \text{ValidQuery}_i\right] \le \Pr\left[\text{MACSForge}_{\Pi,\mathcal{R}_i}(n) = 1\right] \le negl(n)$
- $\Pr\left[PrivK^{\boldsymbol{CCA}}_{\Pi_{AE},\mathcal{A}}(n)\right] \le \frac{1}{2} + (q(n)+1) \cdot negl(n)$

思考：如果Enc和Mac**使用了相同的密钥 k**，上述证明的哪一步不成立？

小结 & 期中考试

- 现代密码学基本法则（Sec 1, 3.1, 3.2）
- Hash与MAC的定义与安全模型（4.1, 4.2, 5.1, 5.5）
- **Hash与MAC的逐级构造**（4.3, 4.4, 5.2, 5.3）
- Hash与MAC的攻击与应用（4.5, 5.4, 5.6）



期中考试：

开卷！（课堂内容，**不含课后习题**）

记得带文具和辅助材料（电子设备除外）