



现代密码学

Modern Cryptography

张方国
中山大学计算机学院

Office: Room 305, IM School Building

E-mail: isszhfg@mail.sysu.edu.cn

HomePage: <https://cse.sysu.edu.cn/content/2460>



第21讲 数字签名（一）

第7章 签名方案

- 引言
- 签名方案的安全性要求
- ElGamal签名方案
- ElGamal签名方案的变形：
Schnorr, DSA, ECDSA
- 可证明安全的签名方案
一次签名，全域Hash
- 不可否认签名
- Fail-stop签名



引言

日常生活中的签名；数字签名方案是一种以电子形式存储的消息签名的方法；

- 签署文件。传统签名是所签署文件的物理部分，而数字签名不是文件的物理部分，只是以某种形式“绑”在文件上；
- 签名验证。传统的签名通过比较其他认证的签名来验证当前签名的真伪。数字签名通过一个公开的验证算法对它进行确认，要能够防止伪造签名。
- 副本。数字签名文件的副本与原签名文件相同，而一个副本的手写签名的纸质文章通常与原来的签名文件区分开来。这意味着我们必须防止一个数字签名消息被重复使用。



与手书签字的区别：手书签字是模拟的，且因人而异。数字签字是0和1的数字串，因消息而异。

与消息认证的区别：消息认证使收方能验证消息发送者及所发消息内容是否被窜改过。当收发者之间没有利害冲突时，这对于防止第三者的破坏来说是足够的。但当收者和发者之间有利害冲突时，就无法解决他们之间的纠纷，此时须借助满足前述要求的数字签字技术。

安全的数字签字实现的条件：发方必须向收方提供足够的非保密信息，以便使其能验证消息的签字；但又不能泄露用于产生签字的机密信息，以防止他人伪造签字。任何一种产生签字的算法或函数都应当提供这两种信息，而且从公开的信息很难推测出用于产生签字的机密信息。此外，还有赖于仔细设计的通信协议。



数字签名的定义

定义7.1 一个签名方案是一个满足下列条件的5元组 $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$:

- \mathcal{P} 是由所有可能的消息组成的一个有限集合;
- \mathcal{A} 是由所有可能的签名组成的一个有限集合;
- \mathcal{K} 是密钥空间, 由所有可能的密钥组成的一个有限集合;
- 对每一个 $K \in \mathcal{K}$, 有一个签名算法 $sig_K \in \mathcal{S}$ 和一个相应的验证算法 $ver_K \in \mathcal{V}$ 。对每一个消息 $x \in \mathcal{P}$ 和每一个签名 $y \in \mathcal{A}$, 每个 $sig_K : \mathcal{P} \rightarrow \mathcal{A}$ 和 $ver_K : \mathcal{P} \times \mathcal{A} \rightarrow \{\text{true}, \text{false}\}$ 都是满足下列条件的函数, (x, y) 为签名消息;

$$ver_K(x, y) = \begin{cases} \text{true} & y = sig_K(x) \\ \text{false} & y \neq sig_K(x) \end{cases}$$



数字签名的定义

- 对每一个 $K \in \mathcal{K}$, sig_K 和 ver_K 应该都是多项式时间函数。
- ver_K 是公开函数, 而 sig_K 是保密的。
- 除了Alice之外, 任何人计算使得 $ver_K(x, y) = true$ 的签名y应该是计算上不可行的。
- 如果Oscar能够计算出使得 $ver_K(x, y) = true$ 的数据对 (x, y) , 而 x 没有事先被Alice签名, 则签名y称为伪造签名。非正式地说, 一个伪造的签名是由Alice之外的其他人产生的有效数字签名。



引言

密码体制7.1 RSA数字签名方案

设 $n = pq$, 其中 p 和 q 是素数。设 $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$, 并定义

$$\mathcal{K} = \{(n, p, q, a, b) : n = pq, ab \equiv 1 \pmod{\phi(n)}\}$$

值 n 和 b 为公钥, 值 p, q 和 a 为私钥。对 $K = (n, p, q, a, b)$, 定义 $sig_K(x) = x^a \pmod{n}$ 以及 $ver_K(x, y) = true \iff x \equiv y^b \pmod{n}$, 其中 $x, y \in \mathbb{Z}_n$ 。

RSA签名体制是不安全的;

阻止这种攻击的一种方法是让消息包含足够的冗余, 使得用这种方法获得的伪造签名对应一个有意义的消息 x 的概率非常小。

Hash函数与数字签名结合使用能阻止这种伪造。



签名与加密

先签名，后加密：给定明文 x , Alice计算她的签名 $y = \text{sig}_{Alice}(x)$, 然后使用Bob的公开加密函数 e_{Bob} 加密 x 和 y , 获得 $z = e_{Bob}(x, y)$ 。密文 z 被传送给Bob。当Bob接收到 z 后, 首先使用解密函数 d_{Bob} 获得 (x, y) , 然后使用Alice的公开验证函数来验证 $\text{ver}_{Alice}(x, y) = \text{true}$ 。

先加密，后签名:即Alice计算 $z = e_{Bob}(x)$ 和 $y = \text{sig}_{Alice}(z)$, Alice将把 (z, y) 发送给Bob, Bob解密 z , 获得 x , 然后用 ver_{Alice} 来验证 z 的签名 y 。

先加密，后签名存在危险, 即如果Oscar获得 (z, y) 对, 他能够用他自己的签名 $y' = \text{sig}_{Oscar}(z)$ 来替换Alice的签名。

注意: Oscar即使在不知道明文 x 的情况下也能对密文 $z = e_{Bob}(x)$ 签名。

然后, 如果Oscar将 (z, y') 发送给Bob, Bob将用 ver_{Oscar} 的签名, Bob可能由此推断明文 x 来自Oscar。

建议, 先签名, 后加密。



签名方案的安全需求

最强的攻击手段和最弱的目标！

我们主要讨论什么是安全的签名方案。如果密码体制一样，我们需要确定一个攻击模型，攻击者的目标以及所提供的安全性类型。

攻击模型：

- 1: 唯密钥攻击。Oscar拥有Alice的公钥，即验证函数 ver_K 。
- 2: 已知消息攻击。Oscar拥有一系列以前由Alice签名的消息，例如 $(x_1, y_1), (x_2, y_2), \dots$ ，其中 x_i 是消息而 y_i 是Alice对这些消息的签名。
- 3: 选择消息攻击。Oscar请求对Alice对一个消息列表签名。因此，他选择消息 x_1, x_2, \dots ，并且Alice提供对这些消息的签名 y_1, y_2, \dots 。



签名方案的安全需求

攻击目标：

- 1：完全破译。攻击者能确定Alice的私钥，即签名函数 sig_K 。因此，他能够对任何消息产生有效的签名。
- 2：选择性伪造。攻击者能以一个不可忽略的概率对另外某个人选择的消息产生一个有效的签名。换句话说，如果给攻击者一个消息 x ，那么他能决定签名 y ，使得 $ver_K(x, y) = true$ 。该消息 x 不是Alice曾经签名过的消息。
- 3：存在性伪造。攻击者至少能为一则消息产生一个有效的签名。换句话说，攻击者能产生一个对 (x, y) ，其中 x 是消息而 $ver_K(x, y) = true$ 。该消息 x 不是Alice曾经签名过的消息。

一个签名方案不可能是无条件安全的。

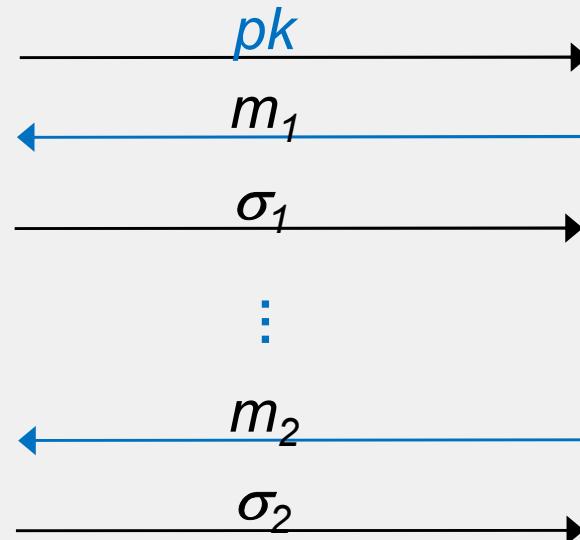
目标是找到计算上或可证明安全的签名方案。



Security of Digital Signatures

- ◆ Existential Unforgeability against Adaptive Chosen-message Attacks (EUF-CMA)

Challenger



Adversary



- **Output:** (m^*, σ^*)
- **Goal:** $\text{Vrfy}_{pk}(m^*, \sigma^*) = 1$

Suc_A^{Unf} is the successful probability of the PPT adversary in winning above Game. The EUF-CMA of Digital Signatures require that Suc_A^{Unf} is a negligible function of the security parameter λ .



先Hash后签名

签名方案几乎总是和一种非常快的公开密码Hash函数结合使用。

假设Alice要对消息 x 签名，这是一个任意长度的比特串。她首先生成消息摘要 $z = h(x)$ ，然后计算 z 的签名，即 $y = \text{sig}_K(z)$ 。然后她将有序对 (x, y) 在信道上传输。

验证者首先通过公开Hash函数 h 重构消息摘要 $z = h(x)$ ，然后检查 $\text{ver}_K(z, y) = \text{true}$ 。

Hash函数 h 的使用并没有削弱签名方案的安全性，因为签名的是消息摘要而非消息本身。有必要使 h 满足一定的属性以便阻止各种各样的攻击。

攻击1

Oscar的一个最显然的攻击是：从一个有效的签名消息 (x, y) 开始，其中 $y = \text{sig}_K(h(x))$ 。然后他计算 $z = h(x)$ 并企图找到 $x' \neq x$ 使得 $h(x') = h(x)$ 。如果Oscar能做到这一点， (x', y) 将成为一个有效的签名消息；因此 y 是消息 x' 的伪造签名。

这是一种使用已知消息攻击的存在性伪造。为了阻止这种攻击，要求 h 是二次原像稳固的。



先Hash后签名

攻击2

Oscar首先找到 $x' \neq x$ 使得 $h(x') = h(x)$ 。然后他将消息 x 发送给Alice，并让Alice对消息摘要 $h(x)$ 签名获得 y 。那么 (x', y) 是有效的签名消息，而 y 是消息 x' 的伪造签名。

这是一种使用选择消息攻击的存在性伪造。为了阻止这种攻击，要求 h 是碰撞稳固的。

攻击3

对一个随机的消息摘要 z 伪造签名对某些签名方案是可能的。也就是，假定签名方案受到使用唯密钥攻击的存在性伪造。

假定Oscar要计算某个消息摘要 z 的签名，然后他找到一个消息 x 使得 $z = h(x)$ 。如果他能做到这一点，那么 (x, y) 是有效的签名消息而 y 是 x 的伪造签名。

这种伪造是该签名方案受到使用唯密钥攻击的存在性伪造。为了阻止这种攻击，要求 h 是原像稳固的。



ElGamal 签名方案

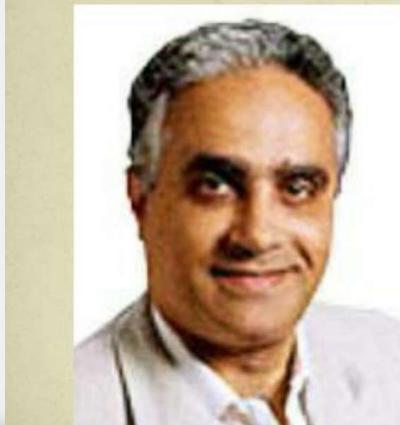
ElGamal签名方案(1985)，该方案的变形已经成为美国国家标准技术研究所采纳为数字签名算法(DSA)。DSA同时吸收了被称为Schnorr签名方案的一些设计思想。

RSA密码体制可以用于加密，又可用于签名。但ElGamal和Schnorr方案都是为签名的目的而专门设计的。

ElGamal签名方案是非确定性的。这意味着对任何给定的消息有许多有效的签名，并且验证算法能够将它们中的任何一个作为真实的签名而接受。

T.ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms[J]. IEEE Transactions on Information Theory, 1985, 31:469-472

TAHER ELGAMAL



- Chief technical officer and Co-Chair of the Board of Directors at Securify (born August 18, 1955).
- Creator of the Elgamal cryptosystem.
- Source:
http://en.wikipedia.org/wiki/Taher_Elgamal



ElGamal 签名方案

密码体制7.2 ElGamal签名方案

设 p 是一个使得在 \mathbb{Z}_p 上的离散对数问题是难处理的素数，设 $\alpha \in \mathbb{Z}_p^*$ 是一个本原元。设 $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{A} = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$, 定义

$$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}$$

值 p, α, β 是公钥, a 是私钥。

对 $K = \{(p, \alpha, a, \beta)\}$ 和一个随机数 $k \in \mathbb{Z}_{p-1}^*$, 定义

$$sig_K(x, k) = (\gamma, \delta)$$

其中

$$\gamma = \alpha^k \pmod{p}, \delta = (x - a\gamma)k^{-1} \pmod{p-1}$$

对 $x, \gamma \in \mathbb{Z}_p^*$ 和 $\delta \in \mathbb{Z}_{p-1}$, 定义

$$ver_K(x, (\gamma, \delta)) = true \leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{p}$$



ElGamal 签名方案的小例子

例7.1 假定选取 $p = 467, \alpha = 2, a = 127$; 那么

$$\beta = \alpha^a \pmod{p} = 2^{127} \pmod{467} = 132$$

若 Alice 要对消息 $x = 100$ 签名，她选取随机数 $k = 213$ (注意 $\gcd(213, 466) = 1$ 且 $213^{-1} \pmod{466} = 431$)。那么

$$\gamma = 2^{213} \pmod{467} = 29$$

并且

$$\delta = (100 - 127 \times 29)431 \pmod{466} = 51$$

任何人通过计算

$$132^{29} \times 29^{51} \equiv 189 \pmod{467}$$

和

$$2^{100} \equiv 189 \pmod{467}$$

来验证这个签名。因此，该签名是有效的。



ElGamal 签名方案的安全性

假定Oscar在不知道 a 的情况下想对给定的消息 x 伪造签名。

如果Oscar选择一个值 γ , 然后试图找出相应的 δ , 那么它必须计算离散对数 $\log_{\gamma} \alpha^x \beta^{-y}$ 。

如果他首先选择 δ , 然后试图找到 γ , 那么他必须“求解”等式

$$\beta^{\gamma} \gamma^{\delta} \equiv \alpha^x \pmod{p}$$

以便获得这个未知的 γ , 这是一个还没有已知的可行的办法来求解的问题。然而, 它与像离散对数问题这样研究的比较透彻的问题似乎没有关系。也许仍然存在某种方法可同时计算 γ 和 δ , 使 (γ, δ) 是一个签名。但是, 没有人发现求解这个问题的方法, 也没有人能够证明不能求解这个问题。

如果Oscar先选择 γ 和 δ , 然后去解 x , 那么他又一次面临着求解离散对数问题的一个实例, 也就是计算 $\log_{\alpha} \beta^{\gamma} \gamma^{\delta}$ 。



ElGamal 签名方案的安全性

同时选择 γ, δ, x 能对任意的消息签名，因此，在唯密钥攻击的情况下进行存在性伪造还是可能的。

已知消息攻击的存在性伪造。

这两种方法都是存在性伪造，但他们似乎还不能被修改成选择性伪造。因此，在使用Hash函数的情况下，这两种方法似乎对ElGamal签名方案的安全性不构成威胁。

随机值 k 不能泄露。

对不同的消息签名时，不能使用相同的 k 。



ElGamal 签名方案的变形

一个1024比特的模导致ElGamal签名有2048比特。对其中许多包括智能卡使用的潜在应用而言，需要的是短的签名。

在1989年，Schnorr提出了一种可看做是ElGamal签名方案的变形的一种签名方案，其签名的长度被大大缩短了。

数字签名算法(DSA)是ElGamal签名方案的另一种变形，它吸收了Schnorr签名方案的一些设计思想。DSA于1994年5月19日发表在Federal Register上，1994年12月1日采纳为标准。

Claus-Peter Schnorr. Efficient Signature Generation by Smart Cards. J. Cryptology, 4(3):161–174, 1991.





Schnorr 签名

密码体制7.3 Schnorr签名方案

设 p 是使得 \mathbb{Z}_p^* 上离散对数问题难处理的一个素数, q 是能被 $p - 1$ 整除的素数。设 $\alpha \in \mathbb{Z}_p^*$ 是1模 p 的 q 次根, $\mathcal{P} = \{0, 1\}^*$, $\mathcal{A} = \mathbb{Z}_p \times \mathbb{Z}_p$, 并定义

$$\mathcal{K} = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}$$

其中 $0 \leq a \leq q - 1$, 值 p, q, α, β 是公钥, a 为私钥。最后, 设 $h : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ 是一个安全Hash函数。

对于 $K = (p, q, \alpha, a, \beta)$ 和一个秘密的随机数 k , $1 \leq k \leq q - 1$, 定义

$$sig_K(x, k) = (\gamma, \delta)$$

其中 $\gamma = h(x || \alpha^k \pmod{p})$ 且 $\delta = k + a\gamma \pmod{q}$ 。

对于 $x \in \{0, 1\}^*$ 和 $\gamma, \delta \in \mathbb{Z}_q$, 验证时通过下面的计算完成的:

$$ver_K(x, (\gamma, \delta)) = true \Leftrightarrow h(x || \alpha^\delta \beta^{-\gamma} \pmod{p}) = \gamma$$





Schnorr 签名

例7.3 假设取 $q = 101, p = 78q + 1 = 7879$ 。3是 \mathbb{Z}_{7879}^* 中的一个本原元，因此取

$$\alpha = 3^{78} \pmod{7879} = 170$$

α 是1模 p 的 q 次根。假设 $a = 75$ ；那么 $\beta = \alpha^a \pmod{7879} = 4567$ 。

现在，假定Alice要对消息 x 签名，她随机选择 $k = 50$ ，并计算 $\alpha^k \pmod{p} = 170^{50} \pmod{7879} = 2518$ ，下一步计算 $h(x||2518)$ ，其中 h 是给定的Hash函数，2518以二进制的形式表示。为了便于解释假设 $h(x||2518) = 96$ ，那么 δ 的计算结果为 $\delta = 50 + 75 \times 96 \pmod{101} = 79$ ，因此，签名为(96,79)。

通过计算 $170^{79} \times 4567^{-96} \pmod{7879} = 2518$ ，并检查 $h(x||2518) = 96$ ，该签名即可得到验证。



DSA

DSS (Digital Signature Standard) 签字标准是1991年8月由美国NIST公布、1994年5月19日正式公布，1994年12月1日正式采用的美国联邦信息处理标准。其中，采用了SHA，其安全性基于解离散对数困难性，它是在ElGamal和Schnorr (1991)两个方案基础上设计的DSS中所采用的算法简记为DSA(Digital Signature Algorithm)。此算法由D. W. Kravitz设计

介绍DSA规范中对ElGamal签名方案验证函数所做的修改。

与Schnorr签名方案一样，DSA使用了 \mathbb{Z}_p^* 的一个 q 阶子群。

在DSA中，要求 q 是长为160比特的素数， p 是长为 L 比特的素数，其中 $L \equiv 0 \pmod{64}$ 且 $512 \leq L \leq 1024$ 。

DSA中的密钥与Schnorr签名方案中的密钥具有相同的形式。

DSA同时还规定了在消息被签名之前，要用SHA-1算法将消息压缩。结果是160比特的消息摘要有320比特的签名，并且计算式在 \mathbb{Z}_p 和 \mathbb{Z}_q 上进行的。



数字签名算法DSA

设 p 是长为 L 比特的素数，在 \mathbb{Z}_p 上其离散对数问题是难处理的，其中 $L \equiv 0 \pmod{64}$ 且 $512 \leq L \leq 1024$ ， q 是能被 $p - 1$ 整除的160比特的素数。设 $\alpha \in \mathbb{Z}_p^*$ 是1模 p 的 q 次根。设 $P = \{0, 1\}^*$, $A = \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, 并定义

$$K = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}$$

其中 $0 \leq a \leq q - 1$ 。值 p, q, α 和 β 是公钥， a 是私钥。

对于 $K = (p, q, \alpha, a, \beta)$ 和一个（秘密的）随机数 k , $1 \leq k \leq q - 1$, 定义

$$sig_K(x, k) = (\gamma, \delta)$$

其中 $\gamma = (\alpha^k \pmod{p}) \pmod{q}$, $\delta = (SHA-1(x) + a\gamma)k^{-1} \pmod{q}$ (如果 $\gamma = 0$ 或 $\delta = 0$, 应该为 k 另选一个随机数)。

对于 $x \in \{0, 1\}^*$ 和 $\gamma, \delta \in \mathbb{Z}_q^*$, 验证是通过下面的计算完成的:

$$e_1 = SHA-1(x)\delta^{-1} \pmod{q}$$

$$e_2 = \gamma\delta^{-1} \pmod{q}$$

$$ver_K(x, (\gamma, \delta)) = true \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \pmod{p}) \pmod{q} = \gamma$$



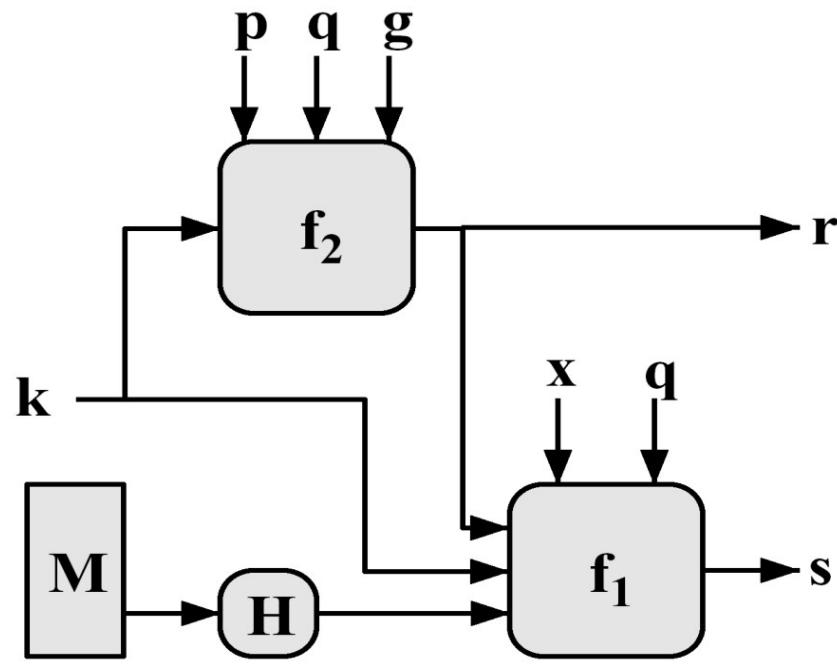
DSA

2001年10月，NIST建议 p 选为1024比特的素数（即 L 的唯一允许值为1024）。这“既不是标准，也不是指南”，但确实表示了对离散对数问题安全性的一些担心。

注意，如果Alice在DSA签名算法中计算出 $\delta \equiv 0 \pmod{p}$ ，她应该放弃 δ ，选择一个新的随机数 k 来构造一个新的签名。我们应该指出，发生 $\delta \equiv 0 \pmod{p}$ 的概率大约是 2^{-160} 。



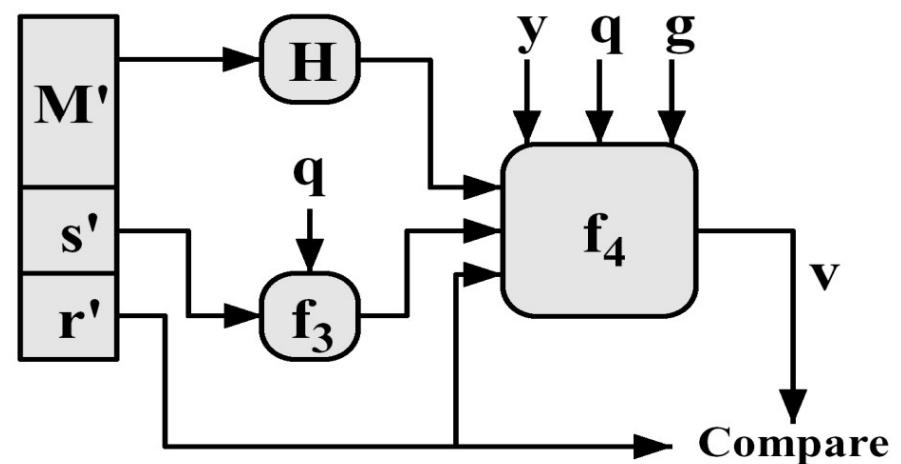
DSA



$$s = f_1(H(M), k, x, r, q) = (k^{-1} (H(M) + xr)) \bmod q$$

$$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$$

(a) Signing



$$w = f_3(s', q) = (s')^{-1} \bmod q$$

$$v = f_4(y, q, g, H(M'), w, r')$$

$$= ((g(H(M'))w) \bmod q) y^{r'} w \bmod q \bmod p \bmod q$$

(b) Verifying