受限伪随机函数：定义的讨论和可验证性构造

咎瑶[1,3]，李红达[1,3]，梁蓓[2]，孟宪宁[1,3]

[1]中国科学院信息工程研究所网络空间安全防御重点实验室 北京中国 100085

[2]北京雁栖湖应用数学研究院 北京中国 101408

[3]中国科学院大学 网络空间安全学院 北京中国 100049

摘要 伪随机函数（Pseudorandom Function, PRF）是现代密码学的基本原语之一，由私钥空间、定义域空间及值域空间所表达。一方从私钥空间随机选取一个私钥，对定义域空间内的任一点，可计算一个 PRF 输出。为了满足日益丰富的安全或应用需求，学者们开启了对 PRF 的扩展性研究，即在 PRF 的基础上增加一些额外的特性，本文着重于 PRF 受限性研究（Constrained PRF，CPRF）。拥有 PRF 私钥的一方，可对定义域空间的某些子集生成受限密钥，该受限密钥可被授权给第三方以计算子集内的所有点处的 PRF 输出！

具体来讲，本文的研究分为两个方面。首先从受限集合的类别、正确性、安全性和额外的属性等四个方面对 CPRF 的定义进行讨论，着重回答目前存在的一些争议点的问题，比如能否用 CPRF 取代函数伪随机函数，单挑战安全性是否等价于多挑战安全性等。其次研究 CPRF 的可验证性(Constrained Verifiable Random Function, CVRF)，提出一个半动态安全的 CVRF 构造。值得注意的是，已知的 CVRF 构造，要么满足弱安全性，如选择挑战安全性；要么满足稍强安全性，如半动态安全性或者动态安全性，遗憾的是，满足稍强安全性的方案皆为非紧致的安全性归约证明，而且支持相对有限的受限集合类。本文的构造在满足稍强安全性，即半动态安全性的同时，不仅具有紧致的安全性归约证明，还支持任意有效可表达的受限集合。

关键词 伪随机函数；受限伪随机函数；受限可验证伪随机函数；安全性归约证明
中图法分类号 TP309.7

Constrained Pseudorandom Functions: Discussion of Definitions and A Verifiability Construction

Zan Yao[1,3], Li Hongda[1,3], Liang Bei[2], Meng Xianning[1,3]

[1] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, CAS, Beijing 100085, China
[2] Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Beijing 101408, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Pseudorandom Function (PRF) is one of the basic primitives in modern cryptography, is represented by a key space, a domain space and a range space. One party selects a secret key randomly from the key space, and calculates a PRF output for any point in the domain. In order to meet the increasingly rich security or application requirements, scholars have started to study the extensibility of PRF, that is, adding some additional features on the basis of PRF. In this paper, we focus on the study of Constrained PRF (CPRF), that is, a party in possession of a PRF secret key can generate a constrained key for some subset of the domain, which can be authorized to a third party to compute the PRF output at all points in the subset!

Specifically, our research is divided into two aspects. Firstly, the definition of CPRF is discussed from four aspects: the constraint category, correctness, security and additional properties, and some controversial issues are emphatically answered, such as whether the Functional PRF can be replaced by CPRF, and whether the security of one challenge is equivalent to the security of multiple challenges. Secondly, the Constrained Verifiable Random Function (CVRF) is studied, and a semi-adaptively secure construction of CVRF is proposed. It is worth noting that the known CVRF constructions, either satisfy weak security such as the selective-challenge security; Either it satisfies stronger security, such as semi-adaptive security or adaptive security. Unfortunately, the proofs satisfying the stronger security are all non-compact reductions and support relatively limited classes of constrained sets. While the construction in this paper not only satisfies the stronger security, that is,

通讯作者：李红达，博士，研究员，Email: lihongda@iie.ac.cn。

本课题得到北京市自然科学基金（项目号：M22001）和国家重点研发计划（项目号：2021YFB2701300）资助。
收稿日期：201X-X-X；修改日期：201X-X-X；定稿日期：201X-X-X

semi-adaptive security, but also has compact security reduction proofs and supports any effectively represented constrained sets.

Key words Pseudorandom Function; Constrained Pseudorandom Function; Constrained Verifiable Random Function; Security reduction proof

1 绪论

伪随机函数(Pseudorandom Function, PRF) 是现代密码学的基本原语之一，由 Goldreich，Goldwasser 和 Micali 等人[1]于 1986 年提出。一个标准的 PRF F 被表述为 $F: K\times X\rightarrow Y$, 其中 K 是私钥空间，X 是定义域空间，Y 是值域空间。拥有私钥 $k \in K$ 的一方，对于 X 空间内的任一点 x，可计算一个 PRF 输出 $y = F(k, x)$。PRF 的安全性要求任意概率多项式时间（Probability Polynomial Time, PPT）的敌手，即使能以黑盒的方式动态地访问 PRF $F(k, \cdot)$并选择一个自己未询问过的挑战点$x^* \in X$，敌手仍无法以一个显著的优势将PRF输出$y = F(k, x^*)$和一个随机函数 $R:X\rightarrow Y$ 的输出 $R(x^*)$区分开来。PRF 被广泛应用于消息认证、对称加密、密钥委托以及学习理论等[2]。

为了满足日益丰富的安全或应用需求，学者们开启了对 PRF 的扩展性研究，即在标准 PRF 的基础上增加一些额外的特性。例如，1999 年 Micali，Rabin 和 Vadhan[3]在 PRF 的基础上增加了可验证性（Verifiable Random Function，VRF），要求接收者在不拥有 PRF 私钥的情况下可对 PRF 的输出进行正确性验证。同一年，Naor，Pinkas 和 Reingold 提出了密钥同态 PRF [4]，要求对于同一个 PRF $F:K\times X\rightarrow Y$ 和定义域空间内的任一点 $x \in X$，计算者可以选择两个不同的私钥 $s \in K$ 和 $t \in K$，然后通过计算 $F(s, x)$ 和 $F(t, x)$得到值 $F(s + t, x)$。2017 年，Boneh，Kim 和 Wu 提出可逆 PRF [5], 要求存在针对 PRF F 的求逆算法$F^{-1}:K\times Y\rightarrow X$，使得任意值 $y \in Y$ 都可通过F^{-1}算法得到其原像 $x \in X$。除此之外，还有分布式 PRF [4]和遗忘 PRF[6-8]等……

本文所研究的 PRF 受限性指的是 PRF 私钥持有者可将一部分计算能力受限制地委托给第三方。具体来讲，2013 年，文献 [9]引入了受限 PRF (Constrained Pseudorandom Function, CPRF)的概念，对于 PRF F 定义域空间 X 上的某个子集 S，PRF 私钥持有者可生成与该子集有关的受限密钥k_S并将其授权给第三方，第三方利用该密钥可以计算子集内的所有点处的 PRF 输出，而子集外的值对第三方仍保持伪随机性！同一时期，Boyle, Goldwasser 和 Ivan 提出函数 PRF（Functional PRF, F-PRF）[10]，Kiayias 等人提出可委托 PRF [11]，两者都与 CPRF 有着极大的相关性。F-PRF 将受限集合 S 替换为函数 $g:X_1\rightarrow X$，这里X_1为函数 g 的定义域空间空间，X 既为函数 g 的值域空间又为 PRF F 的定义域空间。F-PRF 允许 PRF 私钥持有者针对函数 g 生成一个函数密钥 sk_g，任意拥有 sk_g的一方可计算所有 PRF 输出 $y = F(k, x)$，前提是 x 满足:

存在 $z \in X_1$使得 $x = g(z) \in X$。

可委托 PRF 将受限集合 S 替换为一个谓词 $P:X\rightarrow\{0,1\}$, PRF 私钥持有者可以生成相应的谓词密钥, 该密钥可被用于计算所有的 PRF 输出，只要输入 x 满足 $x \in X \wedge P(x) = 1$。此外，与 CPRF 相关的原语还包括可穿孔 PRF[12], 其将受限集合替换为定义域空间内的一个点$x^+ \in X$，该点也被称为穿孔点，即受限密钥可计算除穿孔点外的所有点处的 PRF 输出。

目前 CPRF 相关原语种类繁多，命名交错，使用混乱，因此详细讨论 CPRF 的定义尤为必要。尤其是梳理相似原语的关系, 比如 CPRF 是否与 F-PRF 等价等; 亦或是解决目前关于定义的歧义点, 比如在 CPRF 的安全性定义中，允许敌手进行一次挑战询问是否等价于敌手的多次挑战询问，若等价，归约如何完成等……本文将在第 3 章详细讨论上述问题。

在深刻理解 CPRF 定义的基础上，本文继续对 CPRF 可验证性进行研究。与 VRF 的出现类似，为了使 CPRF 的输出可验证，Fuchsbauer 在 2014 年引入了受限可验证随机函数(Constrained verifiable Random Function, CVRF) [13]。对于每个 CPRF 输出，无论其是由受限密钥k_S还是由私钥 $k \in K$ 计算，计算者都必须提供一个非交互式证明。任何人都可利用证明和公钥来验证接收到的输出的正确性。非正式地，如果一个函数簇 $F: K\times X\rightarrow Y$ 被称为 CVRF，那么一定存在一对公私钥对(pk, sk)，当给定私钥 sk，

1. 求值算法 Eval：对于每个输入 $x \in X$，输出一个伪随机值 $y = F(sk, x) \in Y$ 以及一个相应的证明π。

2. 受限算法 Cons：对于任意子集 $S \subseteq X$，输出一个受限密钥 sk_S，该密钥可被用于计算伪随机值 $y = F(sk, x)$以及相应的证明π，当且仅当 $S(x) = 1$。而满足 $S(x) \neq 1$ 的 x 处的值对受限密钥持有者来说仍然保持伪随机。通常，该受限子集可以是任何有效可表达的集合。

当给定公钥 pk、定义域内的某个点 $x \in X$、以及

相对应的值 $y \in Y$ 和证明 π，验证算法可以验证 $y = F(sk,x)$ 是否成立。CVRF 很好地结合了伪随机性、可验证性和受限委托性，因此可以在需要这些属性的场景中使用。例如，在区块链中，一些分布式共识协议在选择会议领导人时需要不可预测性和可验证性[14]。在微支付协议中，为了降低银行的密钥管理成本，需要受限委托的性质；而想要公平地选择哪一个用户可以进行大额支付又需要不可预测性和可验证性[15]。此外，一个针对受限集合 S 的 CVRF 方案同时意味着一个针对 S 的功能签名方案[16]。受限集合 S 也可以个性化为属性集、标识集和条件运算符等。

然而，目前大多数 CVRF 构造只实现了较弱的安全性[13,16-21]，即选择挑战安全性。在选择挑战安全实验中，PPT 的敌手 A 必须在实验开始时就声明一个挑战点 $x^* \in X$。然后 A 在获得公钥后，可以动态的选择输入点 $x_1,\cdots,x_n \in X$ 向挑战者进行多次求值询问，从而获得一组 PRF 的输出和证明 $(y_1, \pi_1),\cdots(y_n,\pi_n)$，这里 n 代表 A 最大的求值询问次数。与此同时，A 可以动态的选择一组受限集合 $S_1,\cdots S_m \subseteq X$ 向挑战者进行受限密钥询问，从而获得一组受限密钥 $sk_{S_1},\cdots sk_{S_m}$，这里 m 代表 A 最大的受限密钥询问次数。最后，A 仍然无法以不可忽略的优势区分函数值 $y^* = F(sk,x^*)$ 与值域空间中随机选取的值 $y' \in Y$。这里的挑战 $x^* \in X$ 不允许出现在 A 之后的求值和受限密钥询问中。显然，A 在声明挑战点之前获得的信息越多，可以抵抗此类敌手的方案就越安全。因为公钥及多次询问所泄漏给敌手的信息是不可预测的，这直接或间接地帮助敌手选择更有利的挑战，从而增加敌手攻击成功的概率。因此，选择挑战安全性过于薄弱。

事实上，文献[15]实现了稍强的安全性，称为半动态安全性。攻击者可以在发起挑战前执行多项式次地求值和受限密钥询问，而在攻击之后获得公钥。然而，在他们的论文中，受限集合 S 只能包含多项式个点，而且他们的安全证明是非紧致的，存在多项式级别的归约损失。目前实现动态安全的 CVRF 方案[13,16]主要借助复杂性猜测技术，该技术会导致指数级别的归约损失，这是难以忍受的。如何给出更优的 CVRF 构造仍是一大难点。

1.1 贡献

本文的贡献分为两部分，第一部分是 CPRF 定义的讨论，第二部分提出一个半动态且紧安全的 CVRF 方案，并给出了详细的安全性证明。

1.1.1 受限伪随机函数定义的讨论：

本文从 CPRF 受限集合的类别、正确性、安全性以及额外的属性四个方面对 CPRF 的定义进行讨论，着重分析目前存在的歧义点，比如 CPRF 能否完全替代 F-PRF？在 CPRF 安全实验中，敌手的一次挑战询问是否等价于多次挑战询问？CPRF 额外的属性，如隐私性、受限隐藏性及碰撞容忍性的概念等。具体来说，我们表明：

（1）CPRF 不能取代 F-PRF，相反，F-PRF 可将 CPRF 作为一个特殊例子；相较于 CPRF，F-PRF 可被用于保护定义域空间 X 的数据。但给定函数 $g:X_1 \to X$ 及任意 $x \in X$，若无法有效判定是否存在 $z \in X_1$ 使得 $g(z) = x$，那么对某个 F-PRF 方案进行安全性证明可能并不容易，其被作为开放性问题提出。

（2）在同等安全性下，敌手进行一次挑战询问和进行多次挑战询问，两者是等价的。

（3）CPRF 额外属性：隐私性和受限隐藏性等价。

1.1.2 受限伪随机函数的可验证性构造

我们注意到 Liang, Li 和 Chang[21]给出了一种基于不可区分混淆[12]的具有选择挑战安全性的 CVRF 通用构造。他们的构造使用了穿孔 PRF[12]，其允许 PRF 计算者生成一个穿孔密钥，该密钥可被用于计算定义域空间内除穿刺点外的所有点处的 PRF 输出。他们的安全性证明表明，如果存在一个敌手 A，其可以成功攻破 CVRF 的某种安全性，那么需要构造一个敌手 B 去攻击穿孔 PRF 的安全性。由于穿孔 PRF 的性质，B 需要在实验开始时就选定一个穿孔点去请求挑战者获得穿孔密钥，从而借助穿孔密钥来回答 A 的所有询问。由于 B 的单个穿孔点事实上等价于其挑战点（因为只有穿孔点处的值 B 无法计算），为了顺利完成归约，该挑战点也需要等于 A 的挑战，这自然导致了 CVRF 的选择挑战安全性。惊喜的是，我们发现如果将穿孔 PRF 替换为更通用的工具，即 CPRF，则可以满足 CVRF 半动态安全性，因为 B 可以利用 CPRF 的受限委托性质在 A 发出挑战询问之前回答其所有的求值和受限密钥询问。

换句话说，我们的构造是对文献[21]的改进，将选择挑战安全性提高到半动态安全性，并极大简化了它们的证明过程。与现有的半动态安全的构造[15]相比，我们的构造是紧归约的，且支持任意可有效

表达的集合。

在本文的构造中，如果 CVRF 定义在函数簇 $F:K\times X\to Y=\{0,1\}^n$ 以及证明空间 $P=\{0,1\}^p$ 上，则 CPRF 定义在函数簇 $F':K\times X\to Y'=\{0,1\}^{n+p}$ 上，其中 n,p 是与安全参数有关的多项式，形如 $\{0,1\}^n$ 的符号代表 n 比特长的二进制字符串。除了 CPRF，我们还需要一个不可区分混淆器和一个承诺方案 Com[22]。承诺方案的消息空间定义为 $Y=\{0,1\}^n$，随机数空间定义为 $P=\{0,1\}^p$。具体构造过程如下：将 CVRF 的私钥设置为 CPRF 的私钥 $k\in K$。给定 k，CVRF 的求值算法和受限算法如下：

1. 对于任意输入 $x\in X$，求值算法 Eval 首先调用 CPRF 函数 F'，即运行 $v=F'(k,x)\in Y'=\{0,1\}^{n+p}$。然后解析 $v=m||r$ 使 m 为 v 的前 n 比特位，r 为 v 的后 p 比特位，即使得 $m\in Y=\{0,1\}^n$，$r\in P=\{0,1\}^p$。最后输出 CVRF 的输出 $y=m$ 和证明 $\pi=r$。

2. 对于任意受限集合 $S\subseteq X$，受限算法 Cons 首先调用 CPRF 的受限算法生成受限密钥 k_S，其次定义电路 $ConK[k_S]$ 并使用混淆器对其混淆，最后将该混淆电路设置为 CVRF 的受限密钥。

电路 $ConK[k_S]$ 定义为：对于任何输入 $x\in X$，调用受限密钥 k_S 计算 CPRF 输出 v：如果 v 等于一个拒绝符号 $\perp$，则电路输出一对值和证明 $(y,\pi)=(\perp,\perp)$；如果 $v\in\{0,1\}^{n+p}$，则电路按照上述方式解析 $v=m||r$ 并输出 $(y=m,\pi=r)$。

3．为了生成公钥 pk，我们首先定义一个电路 PubK[k]：对于任意输入 $x\in X$，它计算输出 $v=F'(k,x)$ 并解析为 $v=m||r$，输出承诺 $c=Com(m;r)$。我们将 PubK[k] 混淆之后的电路设置为公钥。

4. 对于任意一组公钥 pk、输入 $x\in X$、输出 $y\in Y$ 和证明 $\pi\in P$，验证算法检查如果 $Com(y,\pi)=pk(x)$，则输出 1，否则输出 0。

值得注意的是，上述构造生成的公钥和受限密钥都是混淆电路。

1.2 文章结构

第 2 章说明文中所使用的基本符号和 CPRF 的正式定义。第 3 章从受限集合的类别、正确性、安全性和额外属性等四方面对 CPRF 定义进行讨论。第 4 章给出 CVRF 构造及半动态安全性证明。第 5 章对本文进行总结。

2 基本符号及 CPRF 定义

2.1 基本符号

符号 $\lambda\in N$ 表示安全参数。如果对于所有多项式 $poly(\lambda)$ 和所有足够大的安全参数 λ，都有 $|negl(\lambda)|\le 1/poly(\lambda)$，则 $negl(\lambda)$ 被称为可忽略函数。如果 Y 是一个集合，则 $y\leftarrow Y$ 表示从 Y 中随机抽取一个元素并将其赋值给 y。符号 $\perp$ 表示为非法或者拒绝。一个概率多项式时间（Probability Polynomial Time，PPT）的敌手意味着存在一些多项式 $poly(\lambda)$ 和安全参数 λ，使得敌手的运行时间最多为 $poly(\lambda)$。如果 a,b 都为二进制字符串，$a||b$ 指的是 a 和 b 的串联。对于一个集合 S，其补集表示为 $\bar{S}$，其元素个数为 $|S|$。

任意有效可表达的集合[23]。如果存在一个多项式 $poly(\cdot)$，使得集合 S 可被表达为一个电路 C_S，该电路的规模为 $poly(\cdot)$ 并且有：如果 $s\in S$，则 $C_S(s)=1$；如果 $s\notin S$，则 $C_S(s)=0$，此时 S 被称为一个有效可表达的集合。通常来讲，逻辑电路的规模指的是电路中逻辑门的个数。该集合类同样被称为电路类。

伪随机函数 PRF[1-2]。假设 K,X, Y 分别为空间 $\{0,1\}^{p_1},\{0,1\}^{p_2},\{0,1\}^{p_3}$，其中 p_1,p_2,p_3 都是与安全参数有关的多项式。一个函数簇 $F:K\times X\to Y$ 被称为 PRF 如果它满足以下两个特性：

（1）易于计算：给定密钥 $k\in K$ 和输入 $x\in X$，函数值 $y=F(k,x)$ 是易于计算的。

（2）伪随机性：任意 PPT 的区分器在动态的选择 $x_i\in X$ 并询问得到多项式对 $(x_i,F(k,x_i))$ 后，对于其未询问过的挑战点 $x^*\in X$，区分器仍不能有效地将伪随机函数的输出 $F(k,x^*)$ 与随机函数 $R:X\to Y$ 的输出 $R(x^*)$ 区分开来！

归约损失。一般来说，在论证密码方案安全性时，我们构造一个归约，把破坏密码方案的有效敌手 A 转换为解决某个特定底层困难问题的挑战者 B，如大整数分解问题，离散对数问题等。在归约过程中，如果 B 的运行时间和成功概率与敌手 A 的相近，或者相差一个常数因子，我们就称归约是紧的。通常归约构造的敌手 B 的运行时间与敌手 A 的相近 $t_A\approx t_B$，但成功概率会有差距 $\epsilon_B\ge\epsilon_A/Q$。我们称 Q 为归约损失，特别地，只有当 Q 为某个常数时，我们称归约是紧的。在实际应用中，为了使密码方案在理论上达到特定的安全级别，我们按照安全证明来选择参数，如果归约损失很大，那么方案所需的安全参数就越大，相应地效率就会降低。为了使得密码方案有最优的参数选取，我们需要安全证明是紧归约的，密码方案是紧安全的。

2.2 CPRF 定义

定义 1. 受限伪随机函数 CPRF[9]. 设 λ 为安全参

数，$F: K \times X \to Y$ 为一个带密钥的函数簇，其密钥空间为 K，定义域空间空间为 X，值域空间为 Y。$S = \{S_\lambda \subseteq \{2^X\}\}_{\lambda \in N}$为一个受限集合簇。如果 F 被称为 CPRF，则有 $\Pi_{CPRF} = (CPRF.Gen, CPRF.Cons, CPRF.CEval)$：

- $CPRF.Gen(1^\lambda) \to k$：CPRF 生成算法取安全参数$1^\lambda$作为输入，输出一个私钥 $k \in K$。对于任意 $x \in X$，总有 $y = F(k, x)$ 。
- $CPRF.Cons(k,S) \to k_S$：CPRF 受限算法以一个私钥 k 和一个集合 $S \in S_\lambda$作为输入，输出一个受限密钥k_S。
- $CPRF.CEval(k_S,x) \to y$：CPRF 受限求值算法以一个受限密钥 k_S和 $x \in X$ 作为输入，输出 $y \in Y \cup \{\perp\}$。

以上算法满足以下两个性质：

正确性：对于每一个$\lambda \in N$，$k \leftarrow CPRF.Gen(1^\lambda)$，$S \in S_\lambda$，$k_S \leftarrow CPRF.Cons(k, S)$，以及任意的 $x \in X$：

- 当 $S(x) = 1$，$CPRF.CEval(k_S,x) = F(k,x)$；
- 当 $S(x) = 0$, $CPRF.CEval(k_S,x) = \perp$。

伪随机性：对于任意 PPT 的敌手 A，考虑其与挑战者进行如下安全实验：

1) 准备阶段：挑战者运行 $k \leftarrow CPRF.Gen(1^\lambda)$，并初始化一个集合 $G = \emptyset$。
2) 询问阶段：挑战者为 A 提供求值询问和受限密钥询问：
   a) 求值询问：对于 A 的每个输入 $x \in X$，挑战者返回 $y = F(k,x)$，并更新 $G = G \cup \{x\}$。
   b) 受限密钥询问：对于 A 的每个集合 $S \in S_\lambda$，挑战者返回$k_S \leftarrow CPRF.Cons(k, S)$，并更新 $G = G \cup \{x: S(x) = 1\}$。
3) 挑战阶段：挑战者从 A 处收到一个挑战点$x^* \in X$，其计算$y_0^* = F(k,x^*)$，$y_1^* \leftarrow Y$，然后随机抽取一个比特 $t \in \{0,1\}$，最后将y_t^*发送给 A。
4) 猜测阶段：挑战者收到 A 对 t 的猜测 t'。

将上述实验记为 $Expt^A_{CPRF}(\lambda)$，A 被称为可允许的，当且仅当 A 的挑战询问$x^* \notin G$。我们定义 A 的成功优势为 $Adv^A_{CPRF}(\lambda) = \left|Pr\left[Expt^A_{CPRF}(\lambda): t'=t\right] - \frac{1}{2}\right|$，当 $Adv^A_{CPRF}(\lambda) \le negl(\lambda)$时，F 被称为动态安全 CPRF。

3 CPRF 定义的讨论

在本章，我们主要对 CPRF 的定义进行讨论。首先，我们罗列已知的 CPRF 受限集合类别，强调 F-PRF 中存在映射关系的集合类的重要性并分析其与 CPRF 集合类的关系。其次，我们简单给出 CPRF 正确性与穿孔正确性的定义。再者，我们讨论 CPRF 不同的安全性定义，着重证明单挑战安全性和多挑战安全性的等价性以及选择挑战安全性和选择单密钥安全性的关系等。最后，我们讨论 CPRF 额外属性的定义，比如隐私性、受限隐藏性、碰撞容忍和可验证性等。

3.1 受限集合的类别

目前针对不同的集合类，已有相当多的 CPRF 构造。我们参考文献[24]，按照从上到下依次包含的关系，罗列了所有常见的集合类。假设 $F:K \times X \to Y$ 为 CPRF，其私钥为 $k \in K$，输入为二进制字符串 $x \in X$，通常被表达为 $x = x_1 \cdots x_l$，符号 $x|_{l_1}^{l_2} = x_{l_1} \cdots x_{l_2}$。S 指的是受限集合，$k_S$指的是受限密钥，其可以计算所有的 $F(k, x)$当且仅当 $S(x) = 1$。受限集合的类别及相应文献如下：

- 穿孔点[12]：$S_v(x) = 1$ 当且仅当 $x \neq v$，$v \in X$ 被称为穿孔点。
- 前缀固定的集合[10]：$S_v(x) = 1$ 当且仅当 $x|_1^a = v$，这里 $a \le l$ 是固定的，即 x 的前 a 比特固定为 v。
- 左/右固定集合[9]：假设该集合空间为 X×X，对于每一个 $v \in X$，定义左固定集合 $S^w_{left} = \{(v, x): x \in X\}$和右固定集合 $S^w_{right} = \{(x,v): x \in X\}$。
- 比特固定集合[9,24-26]：对于一个向量 $v \in \{0, 1, ?\}^n$，$S_v(x) = 1$ 当且仅当对于每一个 $i \in [l]$都有 $(x_i = v_i) \vee (v_i = *)$。即，$S_v$中每个字符串与 v 的所有不为?位置都匹配，$S_v$固定于 v。
- t-合取范式（t-conjunctive normal form）[26]：假设 NC^0_t为一个电路类，其至多读取 NC^0电路输入的 t 个指标，对于每个电路$C_i \in NC^0_t$，都有 $S(x) = 1$ 当且仅当 $C(x) = \wedge_i C_i(x)$。
- 点乘集合[24]：对于向量 x,v，有$S_v(x) = 1$ 当且仅当$\langle v,x \rangle = 0$。
- 在$\{NC^1, P/poly\}$中的通用电路集合类[9,18-19,24-25,27-37]：假设对于每一个多项式规模的电路 $C \in \{NC^1, P/poly\}$，总是有一个受限密钥k_C，其可以计算 $F(k, x)$当且仅当 $x \in X$ 且 $C(x) = 1$。

3.1.1 CPRF 与 F-PRF 的关系

除了上述提到的集合类外，仍有一类被我们忽略的集合类，在本文中，我们称之为，存在映射关系的集合类 $g:X_1 \to X$，即对于任意的 $x \in X$，总能有效判定是否存在 $z \in X_1$使得 $g(z) = x$。在讨论该集合类

之前，我们先给出 F-PRF 的定义。

定义 2. F-PRF. 一个带密钥的函数簇 $F:K\times X\to Y$ 被称 F-PRF，如果存在以下算法：

- $Gen(1^\lambda)\to(pp, k)$：生成算法以安全参数 1^λ 为输入，输出一个私钥 $k \in K$ 以及一个公共参数 pp。对于定义域空间内的任意一点 $x \in X$，总有 $y = F(k, x)$。
- $KeyGen(k, g)\to k_g$：密钥生成算法以一个私钥 $k \in K$ 和一个函数的描述 $g:X_1\to X$ 为输入，输出一个函数密钥 k_g。
- $Eval(k_g, g, z)$：求值算法以一个函数密钥 k_g，一个函数 g 以及定义域空间内的一个点 $z \in X_1$ 为输入，输出计算结果 $F(k, g(z))$。

以上算法需要满足以下两个性质：

正确性：对于每个函数 $g \in G$，$\forall z \in X_1$，总有：$\forall k \in K, \forall k_g \leftarrow KeyGen(k,g)$，$Eval(k_g, g, z) = F(k, g(z))$。

伪随机性：对于任意 PPT 的敌手 A，考虑其与挑战者进行如下安全实验：

1) 准备阶段：挑战者运行生成算法 $Gen(1^\lambda)\to(pp, k)$，将公共参数 pp 发给敌手。
2) 函数密钥询问阶段：对敌手 A 的每一个函数询问 g，挑战者运行 $k_g \leftarrow KeyGen(k, g)$，并将 k_g 发送给 A。假设 A 总共询问 l 个函数，这里的 l 通常指的是与安全参数有关的多项式。
3) 挑战阶段：对于 A 的每个挑战询问 $x^* \in X$，挑战者随机抽取一比特的 t，当 $t = 0$ 时，挑战者回复 $F(k, x^*)$；当 $t = 1$ 时，挑战者回复 $R(x^*)$，这里 $R:X\to Y$ 是一个真随机函数。
4) 猜测阶段：挑战者收到 A 对 t 的猜测 t'。

将上述实验记为 $Expt^A_{F\text{-}PRF}(\lambda)$，A 被称为可允许的，当且仅当不存在 $i \in [l]$ 以及 z，使得 $g_i(z) = x^*$。我们定义 A 的成功优势为

$$Adv^A_{F\text{-}PRF}(\lambda) = \left|Pr[Expt^A_{F\text{-}PRF}(\lambda): t' = t] - \frac{1}{2}\right|$$

当 $Adv^A_{CPRF}(\lambda) \le negl(\lambda)$ 时，F 被称为动态安全 CPRF。

接下来，我们将详细分析函数类 g 与集合类 S 的关系，也即 F-PRF 与 CPRF 的关系。首先，若函数 g 被定义为 $g(x) = x$ 当且仅当 $S(x) = 1$，这里 S 可以是上述提到的任意的集合类，CPRF 可被看作 F-PRF 的特例，关于这点已有文章进行说明[38]。

若集合 S 被定义为 $S(x) = 1$ 当且仅当 $\exists z$ s.t $g(z) = x$，这里 g 表示为函数，此时针对集合 S 的受限密钥和针对函数 g 的函数密钥，两者在功能上是等价的。虽然集合和函数的互相表达直觉上暗含了 CPRF 与 F-PRF 的等价性，而且目前大多数文献都默认了这一点，但事实并非如此。

用集合 S 表达函数 g 时，为了保证 S 是有效可判定的，那么是否存在 z 使得 $g(z) = x$ 也应有效可判定。若上述条件有效可判定，此时 F-PRF 与 CPRF 所能计算的 X 空间内的点是一致的，但 CPRF 并不能取代 F-PRF。一方面，相较于 CPRF，使用 F-PRF 可以有效地保护原始定义域空间 X 的信息。例如，当利用 CPRF 构建层级系统时，每个员工具有唯一的身份标识 $ID = x \in X$，其拥有访问系统的密钥 $y = F(k, x)$。不同权限的管理者拥有不同的受限密钥 k_S，利用该密钥，每个管理者可以管理权限 S 内的员工 ID 及员工密钥。假设系统拥有 n 层权限，第 0 层为系统中最高级的管理者，其拥有私钥 k；第 $i \le n-1$ 层为系统中第 i 层管理者，其拥有 k_{S_i} 的权限；第 $i + 1$ 层拥有 $k_{S_{i+1}}$ 的权限，其中 $S_{i+1} \subseteq S_i$；第 n 层为员工，其拥有员工 ID 及相应的密钥。若使用 CPRF，低级别的管理者（或者员工）事实上与最高级的管理者共享的是同一输入域 X，管理者（或员工）可以自然的获得同一域上的其他 ID，甚至可以通过 ID 来推测其他领导的具体权限，而这在保密等级较高的系统中是难以忍受的！若使用 F-PRF，员工和低级别的管理者都只拥有同一级别的输入域 X_{1_i}。当层层传递下来，除非他们拥有所有的 $g_1 \dots g_n$，否则是无法对原始空间 X 进行预测的，这很大程度上避免了原始空间的信息泄漏！另一方面，给定一个 CPRF 方案暗含给定集合类 S，一个给定的集合类 S 并不能囊括任意值域有效可判定的函数类。

上述讨论的前提只是函数 g 值域有效可判定的情况。若不可有效判定时，情况又是怎样的呢？事实上，对于这样的函数 g 和任意的 $x \in X$ 判定是否存在 $z \in X_1$ 使得 $g(z) = x$ 是一个 NP-困难问题。观察 F-PRF 的安全性定义可以发现，为了防止敌手可以平凡的攻击成功，敌手发起的挑战询问 x^* 应是没被询问过的点，因此挑战者需要对敌手询问过的每个函数 g 进行判定，即是否存在 $z \in X_1$ 使得 $g(z) = x^*$。而当函数值域不可有效判定时，就导致一个低效的挑战者。

尽管在安全性定义中，低效的挑战者也是被允许的。然而问题在于，在对某个具体的 F-PRF 方案进行安全性归约证明时，低效的挑战者可能导致无法完成有效归约。具体来讲，为了证明某个方案满足 F-PRF 的伪随机性，往往需要先假设存在一个敌手，其能以不可忽略的优势成功攻击 F-PRF 安全性，然后我们构造一个 PPT 的归约算法去攻击底层的困

难假设。在这种情况下，PPT 的归约算法需要模拟 F-PRF 安全实验中低效的挑战者，尽管归约算法可以借助于底层假设下的挑战者来对敌手的挑战x^*进行判定，但该判定本身是 NP 困难问题，因此无法保证归约确实能完成。针对该问题需要具体问题具体分析，因此我们将其作为开放问题欢迎感兴趣者进行后续的研究！

总之，我们有以下结论：（1）CPRF 不能取代 F-PRF，相反，F-PRF 可将 CPRF 作为一个特殊例子。（2）相较于 CPRF，F-PRF 可被用于保护定义域空间数据。(3）当函数 g 值域不可有效判定时，F-PRF 方案的归约需要具体问题具体看待，有待进一步研究。

3.2 正确性

CPRF 的正确性被定义为：受限密钥k_S按照如下规则进行计算，当输入 x 满足 $S(x) = 1$ 时，k_S计算结果等于 $F(k, x)$；当输入 x 满足 $S(x) = 0$ 时，k_S输出一个拒绝符号⊥。穿孔正确性主要来自于穿孔 PRF：受限密钥可以计算除某个点x^+（或者某个集合S^*）外的所有 x 处的输出 $F(k, x)$。穿孔正确性与正确性本质上是等价的，因为对于同一个集合 S，在正确性被满足的情况下得到的受限密钥k_S和在穿孔正确性被满足的情况下针对 S 的补集 $\bar{S}$ 得到的受限密钥$k_{\bar{S}}$，两者的计算结果是相同的。

3.3 安全性

CPRF 的安全性定义在 2.2 节已经给出，其被称为伪随机性：任意 PPT 的敌手可以动态的选择定义域空间空间上的 x 去询问挑战者并获得回复 $F(k,x)$（称为求值询问）；动态的选择集合 S 去询问挑战者并获得相应的受限密钥k_S（称为受限密钥询问）；在挑战阶段，挑战者随机抽取一个比特 b，对于敌手动态选择的挑战点x^*：若x^*在之前的求值询问或者受限密钥询问中被问过，则挑战者回复 $F(k, x^*)$；若未被询问过，当 $b = 0$ 时，挑战者回复$y^* = F(k, x^*)$；当 $b = 1$，挑战者回复$y^* = R(x^*)$，其中 $R:X\to Y$ 是一个真随机函数。之后，挑战者收到敌手猜测的一个比特 b’，若 $b' = b$，则预示着敌手攻击成功；否则敌手攻击失败！CPRF 被称为动态安全的，如果在上述实验中敌手成功的优势是可忽略的！为了避免敌手平凡的攻击成功，挑战询问中至少有一个挑战点x^*没被询问过，且敌手在此之后进行的所有求值和受限密钥询问都不包括该点。

对于同一个集合序列$S_1,\cdots,S_Q$来说（这里的 S 都只包括该集合内的点 x，即 $S(x) = 1$），当正确性或穿孔正确性分别被满足时，敌手可以询问的挑战空间是不同的。具体来讲，当正确性满足时，挑战点$x^* \in \overline{NQ} = S_1 \cup \ldots \cup S_Q$；当穿孔正确性被满足时，$x^* \in NQ = S_1 \cap S_2 \ldots \cap S_Q$，NQ 代表为被询问过的空间，也即挑战空间。显然，为了保证至少存在一个未被询问过的挑战点，有 $NQ \neq \emptyset$。

另外，当敌手发起的挑战点被询问过，挑战者将做出一致的回复，只有当接收到未被询问过的点，挑战者的 F 或 R 回复才有意义，因此在下文中，敌手的所有挑战询问被默认为是未被询问过的点，这样的敌手也被称为可允许的敌手。

3.3.1 多挑战与单挑战的关系

若敌手未被询问过的挑战点的个数为 1 个，该安全性被称为单挑战安全性（2.2 节给出的 CPRF 定义满足该情况）；若个数为多个，则被称为多挑战安全性！我们有以下定理：

定理 1. 多挑战安全性和单挑战安全性等价.

（1）若存在 PPT 的敌手 A 能以不可忽略的优势ϵ成功攻击多挑战安全性且多挑战的询问次数为Q_c次，则可以构造一个敌手 B，其能以ϵ/Q_c的优势成功攻击单挑战安全性。(2）若存在一个 PPT 的敌手 A 能以ϵ的优势攻击单挑战安全性，则可以构造一个敌手 B，其能以ϵ的优势成功攻击多挑战安全性。

证明. 关于（1）。考虑直接的一次归约！若存在一个攻击多挑战安全性的敌手 A，我们需要构造一个敌手 B 去攻击单挑战安全性。在该归约证明过程中，关键性的问题是，敌手 B 只可对挑战者进行一次挑战询问，而面对 A 的多次挑战询问，B 无法成功回答！尽管论文[9,29]中提到，一个满足单挑战安全的 CPRF 方案也满足多挑战安全性，这可以通过一个标准的混杂论证（Hybrid Argument）技术来实现，然而他们没有给出具体的证明。我们接下来给出完整的证明。具体来讲，假设敌手 A 的挑战询问次数为Q_c次，我们定义$Q_c + 1$ 个实验，每个实验中敌手 A 与挑战者挑战阶段的交互过程如下：

实验 0：挑战者对 A 的每个挑战询问都回复 $F(k, x^*)$。

实验 1：与上一个实验相同，除了 A 的第一个挑战询问，挑战者回复 $R(x^*)$，其中 $R:X\to Y$ 是一个真随机函数。

……

实验 i($i < Q_c$)：与上一个实验相同，除了 A 的第 i 个挑战询问，挑战者回复 $R(x^*)$，其中 R:X→Y 是一个真随机函数。

……

实验Q_c：与上一个实验相同，除了 A 的第Q_c个挑战询问，挑战者回复 $R(x^*)$，其中 R:X→Y 是一个真随机函数。

在以上实验中，实验 0 代表 $b = 0$ 的情况；实验Q_c代表 $b = 1$ 的情况，若 A 可以以ϵ的优势成功攻击多挑战安全性，意味者 A 可以区分实验 0 与实验Q_c，那么至少存在一个 $j < Q_c$使得 A 可以以至少ϵ/Q_c的优势区分实验 j 和实验 $j + 1$。此时，我们可以构造一个B_j，使其可以攻击单挑战安全性。具体来说，B_j与实验 j 中的挑战者行为一致，除了按照以下的方式回复 A 的第 i 次挑战询问：

1. 当 $i \le j$ 时，B_j随机选择一个随机函数 $R':X\to Y$ 计算所有的挑战回复。
2. 当 $i = j + 1$ 时，B_j将该挑战询问发送给单挑战安全实验中的挑战者，并获得相应的回复y_b^*，之后B_j将y_b^*发送给 A。
3. 当 $i > j + 1$ 时，B_j触发求值询问并得到单挑战安全实验中的回复 $F(k, x^*)$。

很显然，当 $i = j + 1$ 时，若单挑战实验中挑战者回复B_j时使用的是 PRF F 计算的，则 A 的视图相当于实验 j，若单挑战实验中挑战者使用的是随机函数 R 计算回复的，则 A 的视图相当于实验 $j + 1$。若 A 可以以不可忽略的优势区分实验 j 和实验 $j + 1$，预示B_j可以利用 A 以相同的优势区分单挑战安全实验中的 F 和 R。值得注意的是，尽管B_j计算用的 R'和单挑战安全实验中的 R 不一致，但其都是随机分布，因此从 A 的视图来看，两者是统计不可区分的。挑战者用 R 对第 $j + 1$ 个挑战进行回复时，其计算结果等于B_j曾回复过的点的概率为 $i/2^{|y|}$，这里|y|为二进制比特串 $y \in Y$ 的长度。由于|y|和 i 通常是与安全参数有关的多项式，因此上述概率通常是可忽略的。

以上无论单挑战还是多挑战安全性实验都基于动态安全性实验，因此B_j在将 A 的第 $j + 1$ 个挑战作为单挑战询问发给挑战者后，仍可以进行求值询问！只是若假设 A 成功的优势为ϵ且询问次数为一个多项式Q_c，则B_j成功的优势为$\frac{1}{Q_c}\cdot\epsilon$，即导致一个多项式级别的归约损失。另外，只要是基于同一安全实验，一个满足单挑战安全的 CPRF 方案也满足多挑战安全性。

关于（2）。一个满足多挑战安全性的方案，也一定满足单挑战安全性。如果存在一个攻击单挑战安全性的敌手 A，我们可以构造一个攻击多挑战安全性的敌手 B。当 A 触发了一个挑战询问，则 B 就将该挑战询问发给自己的挑战者并将挑战者的回复发送给 A。B 的其他挑战询问其可以在任意节点触发！在这个归约中，B 成功的优势等于 A 成功的优势，因此该归约是一个紧归约。

由于单挑战者和多挑战安全性等价，在以下安全性讨论中，我们默认敌手的挑战为单挑战。

3.3.2 其他安全性

由于敌手可以发起求值询问、受限密钥询问和挑战询问，在每种安全实验中，敌手发起这三个询问的不同次序导致了不同的安全性。这是因为敌手在发起挑战询问之前，能获得信息越多，则代表敌手的攻击能力越强，那么能抵抗此类攻击的方案的安全性就越高！我们将按照安全性从低到高的顺序对已有的安全性进行罗列：

1. 选择挑战安全性：敌手先选择一个挑战询问x^*，后进行求值询问和受限密钥询问。值得注意的是，敌手的所有求值和受限密钥询问都不能包括x^*。
2. 半动态安全性，也叫弱动态安全性：敌手可以先进行求值询问，后进行挑战询问，之后再进行求值和受限密钥询问。
3. 动态安全性：敌手可以动态的进行求值询问、受限密钥询问和挑战询问。

尽管在动态安全性中，敌手的求值询问、受限密钥询问和挑战询问可以在安全实验的任一节点进行，但定义 2.1 只考虑了挑战之前的求值和受限密钥询问，并未考虑之后的，这主要是因为在挑战之后的询问与选择挑战安全的交互模式相同，而已知方案的动态安全性暗含了选择挑战安全性。

除了以上有明显强弱关系的 CPRF 安全性外，还有选择单密钥安全性（目前已有的定义是在穿孔正确性下定义的[37]）：敌手先进行一次受限密钥询问（即选择一个集合 S 去询问挑战者并获得相应的受限密钥 sk_S），后选择一个挑战点x^*，其满足 $S(x^*) = 1$。目前没有文献将此安全性与以上安全性的关系进行区分，我们经研究发现：

定理 2. 若一个 CPRF 方案满足选择挑战安全性，那么它也一定满足选择单密钥安全性，前提是在选择单密钥安全实验中敌手无法计算的点的个数是多

项式个，即 $NQ = \overline{S}$ 包含多项式个点。

证明. 若存在一个 PPT 的敌手 A 可以以不可忽略的优势ϵ攻击选择单密钥的安全性且 A 单密钥询问的集合 S 满足$|S| = Q$，即 S 内元素的个数为一个多项式 Q，那么我们可以构造一个敌手 B 其能以ϵ/Q 的优势攻击选择挑战安全性。B 的运行过程如下：

1) 在试验一开始时，B 收到来自于 A 的受限集合询问 S。
2) B 随机选取一个点x^+使得 $S(x^+) = 1$，然后将x^+发给挑战者。
3) B 将集合 $\overline{S}$ 发给挑战者，并收到挑战者的回复 sk_S。然后 B 将 sk_S 发送给 A。
4) B 收到 A 的挑战询问x^*，这里 $S(x^*) = 1$：
   - 若$x^+ = x^*$，B 向挑战者请求挑战回复，挑战者随机抽取一个比特 t，设置 $y_0^+ = F(k, x^+)$，$y_1^+ \leftarrow Y$，后将y_t^+发送给 B。B 直接将y_t^+发送给 A。后 B 收到 A 的一比特的猜测并将其设置为 t'。
   - 若$x^+ \neq x^*$，B 停机并输出一个随机比特 t'。

当$x^+ = x^*$时，B 的随机抽样 b 默认为 $b = t$，因此其收到 A 的猜测 b'后也默认$t' = b'$，那么 B 输出$t' = t$ 的概率实际上等于 A 猜测$b' = b$ 的概率。那么敌手 B 在选择挑战安全实验中成功输出$t' = t$ 的优势，即

$$
\begin{aligned}
Adv_B &= \left|Pr\left[Expt_B^{sel\text{-}cha}:t'=t\right] - \frac{1}{2}\right| \\
&= \left|Pr\left[x^+\neq x^*\right]\cdot Pr\left[B:t'=t\right] + Pr\left[x^+=x^*\right]\right. \\
&\qquad \left.\cdot Pr\left[B:t'=t\right] - \frac{1}{2}\right| \\
&= \left|\left(1-\frac{1}{Q}\right)\cdot\frac{1}{2} + \frac{1}{Q}\cdot Pr\left[A:b'=b\right] - \frac{1}{2}\right| \\
&= \left|\frac{1}{Q}\cdot\left(Pr\left[A:b'=b\right] - \frac{1}{2}\right)\right| \\
&= \frac{1}{Q}\cdot Adv_A
\end{aligned}
$$

由于方案满足选择挑战安全性，因此 B 成功的优势 Adv_B是可忽略，那么 A 成功的优势 Adv_A亦是可忽略的。在以上归约证明中，存在多项式级别的归约损失。

目前，支持电路集合的动态安全 CPRF 的构造已经取得了一定的进展。除文献[25,27,37]在随机谕言机模型中实现了 CPRF 外，文献[24]在标准模型中给出了基于 IO 和 shift-hiding 移位函数的 P/Poly 的构造，文献[37]在基于 IO 和子群隐藏假设下实现了 NC^1 的动态单密钥安全 CPRF。

3.4 额外的属性

除了上述提到的动态安全性外，许多文献，在 CPRF 的基础上增加了额外的特性，比如隐私性（Privacy）、受限隐藏性（Constrain- hiding）、碰撞容忍和可验证性等。我们接下来，将分别对这些额外的属性进行总结。

隐私性：是由文献[35]首次提出来的，是指受限密钥 sk_S不会泄漏受限集合 S 的信息。其在文献[34,26]中也被称为受限隐藏性。

碰撞容忍性：是由文献[26]首次提出来的，在安全性定义中，若敌手只可以进行一次受限密钥询问，则称为单密钥安全；若敌手可进行 Q 次受限密钥询问，则称为 Q-碰撞容忍安全。

可验证性：是由文献[13]首次提出，CPRF 计算者不仅需要公布与求值算法有关的公钥信息，而且对每个 CPRF 输出，需要同时提供一个非交互式证明。接收者利用公钥和证明可对接收到的 CPRF 输出的正确性进行验证。关于 CPRF 可验证性的研究，也是本文的另一个研究重点，我们将在下一章节给出正式的定义以及具体的构造。

4 CPRF 的可验证性研究：一个构造

目前，CVRF 的构造已经取得了很大的进展。文献[13,16]基于多线性判定性 Diffie-Hellman 假设构造了选择挑战安全的 CVRF 方案。其他实现选择挑战安全的构造要么基于不可区分混淆[20-21]要么基于函数加密[18]，同时依赖一些额外的假设。文献[15]基于不可区分混淆和可穿刺可验证 PRF 实现了稍强一点的安全性，即半动态的安全性，遗憾的是，其安全证明是非紧致的，即存在多项式级别的归约损失。本文给出的半动态安全性的构造，是从 CPRF 的自然延伸，且存在紧致的安全性证明。

4.1 CVRF 定义

定义 3. 受限可验证伪随机函数(Constrained Verifiable Random Functions, CVRF) [16]. 一个函数簇 $F:K\times X\rightarrow Y$ 可被称为关于集合空间 $S = \{S_\lambda \subseteq \{2^X\}\}_{\lambda\in N}$的 CVRF，若存在受限密钥空间 K'和证明空间 P 和算法 $\Pi_{CVRF} = (Setup, Eval, Cons, CEval, Verify)$。算法描述如下：

- $Setup(1^\lambda)\rightarrow(pk,sk)$：Setup 算法以安全参数$1^\lambda$为输入，输出一对公私钥$(pk,sk)$。
- $Eval(sk,x)\rightarrow(y,\pi)$：求值算法以密钥 sk 和定义域空间空间内的一点 $x \in X$ 作为输入，输出一个值-证明对$(y,\pi) \in Y \cup P$。为了简单起见，该算法

也可以写成值函数 F 和证明函数 P：对于任意 $x \in X$，存在 $y = F(sk,x)$，$\pi = P(sk,x)$ 。

- $Cons(sk,S) \to sk_S$：受限算法以密钥 sk 和集合 $S \in S_\lambda$作为输入，输出一个受限密钥 $sk_S \in K'$。
- $CEval(sk_S,x) \to (y,\pi)$：受限求值算法以一个受限密钥 sk_S和定义域空间内的一个点 x 作为输入，输出一个值-证明对$(y,\pi) \in Y \times P \cup \{(\perp,\perp)\}$。
- $Verify(pk,x,y,\pi) \to 1/0$：验证算法以公钥 pk、输入 $x \in X$、函数值 $y \in Y$、证明$\pi \in P$ 为输入，输出 1/0。

以上算法满足以下性质：

可证明性：对于所有$\lambda \in N$，$(pk, sk) \leftarrow Setup(1^\lambda)$，任意 $S \in S_\lambda$，$sk_S \leftarrow Cons(sk, S)$，$x \in X$，以及$(y,\pi) \leftarrow CEval(sk_S,x)$，

- 若 $S(x) = 1$,则有 $y = F(sk, x)$且 $Verify(pk, x, y,\pi) = 1$。
- 若 $S(x) = 0$，则$(y,\pi) = (\perp,\perp)$。

唯一性：对于所有$(pk,sk) \leftarrow Setup(1^\lambda)$和输入点$x \in X$，不存在$(y_0, y_1, \pi_0, \pi_1)$使得$y_0 \neq y_1 \wedge Verify(pk, x, y_i, \pi_i) = 1, for\ i = [0,1]$。

伪随机性：对于任意 PPT 的敌手 A，考虑 A 与挑战者进行以下的交互实验：

1) 准备阶段：挑战者运行$(pk, sk) \leftarrow Setup(1^\lambda)$并初始化集合 $G = \emptyset$。
2) 询问阶段：挑战者为 A 提供两种类型的询问，分别是求值询问和受限密钥询问。求值询问：对于 A 的每个询问 $x \in X$，挑战者返回$(y,\pi) \leftarrow Eval(sk, x)$并更新 $G = G \cup \{x\}$。受限密钥询问：对于 A 的每个询问 $S \in S_\lambda$，挑战者返回 $sk_S \leftarrow Cons(sk, S)$并更新 $G = G \cup \{x:S(x) = 1\}$。
3) 挑战阶段以及收到 pk：挑战者收到 A 的一个挑战x^*，首先计算$y_0^* = F(k, x^*)$，$y_1^* \leftarrow Y$，后抽取$b \leftarrow \{0,1\}$，然后将 pk 和y_b^*发送给 A。
4) 猜测阶段：挑战者收到 A 对 b 的猜测 b'。

将上述实验记为 $Expt_{CVRF}^{A}(1^\lambda)$，若 A 的挑战满足$x^* \notin G$，则 A 被称为可允许的。我们定义 A 的成功优势为

$$Adv_{CVRF}^{A}(1^\lambda) = \left| Pr\left[Expt_{CVRF}^{A}(1^\lambda) : b' = b\right] - \frac{1}{2} \right|,$$

若 $Adv_{CVRF}^{A}(1^\lambda) \leq negl(\lambda)$，则 F 被称为半动态安全的 CVRF。

4.2 构造所需工具

除了 CPRF 外，我们 CVRF 构造仍需要以下两个工具：不可区分混淆[12,39]和承诺方案[22]。接下来我们将给出正式定义。

4.2.1 不可区分混淆

定义 4. 不可区分混淆[12,39]. 对于电路类 $C = \{C_\lambda \subseteq \{2^X\}\}_{\lambda \in N}$，如果满足以下条件，则 PPT 算法 O 称为一个安全的不可区分混淆器：

完备性：对于任意的安全参数$\lambda \in N$，任意的电路$C \in C_\lambda$及其输入 x，都有

$$Pr[\hat{C}(x) = \hat{C}(x) : C \leftarrow O(\lambda, C)] = 1.$$

不可区分性：对于任意(不一定是一致的)PPT 的敌手 Samp 和 D，存在可忽略函数 negl(λ)，使得对于所有安全参数$\lambda \in N$，如果存在$(C_0, C_1, \sigma) \leftarrow Samp(1^\lambda)$，使$Pr\left[|C_0| = |C_1| \wedge \forall x, C_0(x) = C_1(x)\right] \leq 1 - negl(\lambda)$，这里的σ指的是辅助信息，那么对于带辅助信息σ的敌手 D，总有：

$$Pr\left[D(\sigma, O(\lambda, C_0)) = 1\right] - Pr\left[D(\sigma, O(\lambda, C_1)) = 1\right] \leq negl(\lambda)。$$

4.2.2 非交互承诺方案

非交互承诺由文献[22]引入，其可以由单向函数构造。

定义 5. 非交互承诺[22]. 一个非交互承诺方案由一个多项式时间的承诺算法组成：

- $Com(m;r) \to c$：承诺算法以消息 $m \in \{0,1\}^{p_1}$和随机数 $r \in \{0,1\}^{p_2}$作为输入，输出一个承诺值 $c \in \{0,1\}^{p_3}$，其中p_1，p_2，p_3是与安全参数$\lambda \in N$有关的多项式。

其满足以下性质：

完美绑定性：对于每一个安全参数$\lambda \in N$和字符串$c \in \{0,1\}^{p_3}$，最多存在一个 $m \in \{0,1\}^{p_1}$使得 c 是对消息 m 的承诺。形式上，对于任意$\lambda \in N$，任意$r_0, r_1 \in \{0,1\}^{p_2}$：如果$Com(m_0;r_0) = Com(m_1;r_1)$，则$m_0 = m_1$。

计算隐藏性：对于任意 PPT 的敌手 A 和任意的消息$m_0, m_1 \in \{0,1\}^{p_1}$，有

$$\left| Pr\left[t' = t \;\middle|\; \begin{array}{c} r_0, r_1 \leftarrow \{0,1\}^{p_2}, \\ t \leftarrow \{0,1\}, \\ c_t = Com(m_t; r_t), \\ t' \leftarrow A(m_0, m_1, c_t) \end{array} \right] - \frac{1}{2} \right| \leq negl(\lambda)$$

4.3 构造过程

以下构造主要是从 CPRF 衍生过来的。我们给出了一个 CVRF 通用构造。将 CVRF 定义在函数簇 $F: K \times X \to Y = \{0,1\}^n$和证明空间 $P = \{0,1\}^p$上，其中 n 和 p 是与安全参数$\lambda \in N$有关的多项式，$S = \{S_\lambda \subseteq \{2^X\}_{\lambda \in N}$

为受限集合类。我们需要以下几个工具：

- ✔ CPRF：定义在函数簇 $F':K\times X\to Y'=\{0,1\}^{n+p}$ 上，其包括以下三个算法：Π_{CPRF} = (CPRF.Gen, CPRF.Cons, CPRF.CEval)。
- ✔ Com：非交互承诺方案，其消息空间定义为$\{0,1\}^n$，随机数空间定义为$\{0,1\}^p$。
- ✔ O：安全的不可区分混淆器。

Π_{CVRF} = (Setup, Eval, Cons, CEval, Verify)描述为：

1. Setup(1^λ)：准备算法以安全参数1^λ为输入，然后按照以下步骤运行：
   1) 运行 CPRF.Gen(1^λ)→k，设置私钥 sk = k。
   2) 定义电路 PubK[k]（如图 1），并将其混淆版本设置为公钥，即 pk←O(λ,PubK[k])。

常量：k；输入：x
1）计算 v = F'(k, x)
2)解析 v = m||r 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特；
3）输出一个承诺值 c = Com(m;r)。

图 1　PubK[k]的电路描述

Figure 1 The Description of Circuit PubK[k]

   3) 输出一对公私钥(pk, sk)。
2. Eval(sk, x)：求值算法以私钥 sk = k 和定义域空间内的一点 x ∈ X 为输入，然后按照以下步骤运行：
   1) 计算 v = F'(k, x)并解析 v = m||r 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特。
   2) 输出一对函数值和证明(y, π) = (m, r)。

另外，对于任意输入点 x ∈ X，定义 CVRF 的值函数为 y = F(sk, x) = m，证明函数为π = P(k, x) = r。

1) Cons(sk, S)：受限算法以私钥 sk = k 和一个受限集合 $S\in S_\lambda$为输入，然后按照以下步骤运行：
   a) 计算 CPRF.Cons(k, S)→k_S。
   b) 定义电路 ConK[k_S]（如图 2），并将其混淆后的电路设置为受限密钥，即 $sk_S\leftarrow O(\lambda,ConK[k_S])$，然后输出 sk_S。

常量：k_S；输入：x
计算 v = CPRF.CEval(k_S,x)，然后检查：
1）如果 v ≠⊥，那么解析 v = m||r 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特；然后输出(y, π) = (m, r)。
2）如果 v =⊥，则输出(y, π) = (⊥, ⊥)。

图 2　ConK[k_S]的电路描述

Figure 2 The Description of Circuit ConK[k_S]

2) CEval(k_S,x)：受限求值算法以受限密钥 sk_S和定义域空间空间内的一个点 x ∈ X 为输入，输出一对函数值和证明(y, π) = sk_S(x)。
3) Verify(pk, x, y, \pi) = 1/0：验证算法以公钥 pk，定义域空间空间内的一个点 x ∈ X，函数值 y ∈ Y 和证明π ∈ P 为输入，验证 pk(x) = Com(y;π)是否成立，若成立，则输出 1；否则，输出 0。

注意：以上算法中，公钥 pk 和受限密钥 sk_S都为不可区分混淆器混淆后的电路，也具备电路属性，因此 sk_S(x)和 pk(x)的表述及计算形式也是被允许的。

4.4 安全性证明

定理 3. 若 O 为不可区分混淆器，F 为自适应安全 CVRF，Com 为安全的非交互承诺方案，则上述构造为半动态安全 CVRF。

证明. 上述 CVRF 构造的**可证明性**主要来自于 CPRF 的正确性和混淆器的完备性。

唯一性是从承诺方案的完美绑定性推导出来的。若唯一性不满足，则一定存在一个元组 (pk, x, $(y_0,\pi_0),(y_1,\pi_1)$)，满足 $y_0\neq y_1$且 Verify(pk, x, y_1,π_0) = Verify(pk, x, y_1,π_1) = 1。也即 $y_0\neq y_1$且 pk(x) = Com$(y_0;\pi_0)$ = Com(y_1, π_1)，这与承诺方案的完美绑定性相矛盾，因此假设不成立，即上述方案满足唯一性。

证明**半动态安全性**，我们定义了一个实验序列，其中每个实验是由一个可允许的 PPT 的敌手 A 和一个挑战者完成交互。对于不同的实验 $Expt^i_A$，我们将敌手 A 攻击成功的优势定义为

$$Adv^i_A(\lambda) = |Pr\left[Expt^i_A(\lambda)=1\right] - \frac{1}{2}|,$$

为书写简便，我们有时会将上述等式中的安全参数略写。所有实验的描述如下：

$Expt^0_A$：与定义 3 中半动态安全实验相同：

1. 准备阶段：挑战者运行 k←CPRF.Gen(1^λ)，设置私钥 sk = k，然后初始化一个空集合 G = ∅。
2. 询问阶段：挑战者回答 A 的以下询问：
   - 求值询问：敌手 A 用定义域空间空间内的一个点 x ∈ X 询问挑战者，挑战者首先计算 v = F(k,x)，然后解析 v = m||r 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特，最后返回 (y, π) = (m, r)并更新集合 G = G ∪ {x}。
   - 受限密钥询问：敌手 A 选择一个受限集合 $S\in S_\lambda$询问挑战者，挑战者首先运行 k_S←CPRF.Cons(k, S)，然后定义电路 ConK[k_S]（如图 2），并将其混淆后的版本设置为受限密钥 $sk_S\leftarrow O(\lambda, ConK_{k_S})$，最后返回 sk_S并更新集合 G = G ∪ {x:S(x) = 1}。
3. 挑战和回复 pk 阶段：挑战者收到 A 的一个挑战询

问$x^* \in X$，然后按照以下步骤运行：

1）定义一个电路 PubK[k]（如图 1），并将其混淆后的版本设置为公钥，即 pk←O(λ, PubK[k])。

2） 挑战者先随机选取一个比特 $b \in \{0, 1\}$，若 $b = 0$，挑战者计算$v^* = F(k, x^*)$并解析$v^* = m^*||r^*$使得m^*为v^*的前 n 比特，r^*为v^*的后 p 比特，然后返回$y_0^* = m^*$；若 $b = 1$，挑战者计算$y_1^* \leftarrow \{0, 1\}^n$。

3）最后，挑战者将(pk, y_b^*)发送给 A。

4. 猜测阶段：挑战者收到 A 对 b 的猜测 b′。

$Expt_A^1$：除了对电路 PubK 的描述外，本实验与 **$Expt_A^0$** 相同。具体来讲，当挑战者接收到敌手 A 发出的挑战询问x^*后，先计算$v^* = F(k, x^*)$并解析$v^* = m^*||r^*$使得m^*为v^*的前 n 比特，r^*为v^*的后 p 比特，后生成一个承诺值$c^* = Com(m^*;r^*)$。随后，挑战者定义一个集合$S^* = X/\{x^*\}$并生成受限密钥$CPRF.Cons(k, S^*) \rightarrow k_{S^*}$。最后，挑战者定义一个新的电路$PubK[x^*,c^*,k_{S^*}]$如图 3 所示：

常量：x^*,c^*,k_{S^*}；输入：x 1）如果 $x = x^*$，输出c^* 2）如果 $x \neq x^*$，计算 $v = CPRF.CEval(k_{S^*}, x)$并解析 $v = m\|\|r$ 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特，最后输出 $c = Com(m;r)$。

图 3 $PubK[x^*,c^*,k_{S^*}]$的电路描述

Figure 3 The Description of Circuit $PubK[x^*,c^*,k_{S^*}]$

本实验具体的描述如下：

1. 准备阶段：挑战者运行 $k \leftarrow CPRF.Gen(1^\lambda)$，设置私钥 $sk = k$。然后初始化一个空集合 $G = \emptyset$。

2. 询问阶段：挑战者回答 A 的以下询问：

- 求值询问：敌手 A 用定义域空间空间内的一个点 $x \in X$ 询问挑战者，挑战者首先计算 $v = F(k,x)$，然后解析 $v = m||r$ 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特，最后返回 $(y, \pi) = (m, r)$并更新集合 $G = G \cup \{x\}$。
- 受限密钥询问：敌手 A 选择一个受限集合 $S \in S_\lambda$询问挑战者，挑战者首先运行 $k_S \leftarrow CPRF.Cons(k, S)$，然后定义电路 $ConK[k_S]$（如图 2），并将其混淆后的版本设置为受限密钥 $sk_S \leftarrow O(\lambda, ConK_{k_S})$，最后返回 sk_S并更新集合 $G = G \cup \{x:S(x) = 1\}$。

3. 挑战和回复 pk 阶段：挑战者收到 A 的一个挑战询问$x^* \in X$，然后按照以下步骤运行：

1）挑战者**首先计算**$v^* = F(k, x^*)$并解析$v^* = m^*||r^*$使得m^*为v^*的前 n 比特，r^*为v^*的后 p 比特，然后生成一个承诺值$c^* = Com(m^*;r^*)$；**然后挑战者定义一个集合**$S^* = X/\{x^*\}$并生成相应的受限密钥$CPRF.Cons(k, S^*) \rightarrow k_{S^*}$；**最后，挑战者定义电路** **$PubK[x^*,c^*,k_{S^*}]$**（如图 3）并将其混淆后的版本设置为公钥 $pk \leftarrow O(\lambda, PubK[x^*,c^*,k_{S^*}])$。

2） 挑战者先随机选取一个比特 $b \in \{0, 1\}$，若 $b = 0$，挑战者返回$y^* = m^*$；若 $b = 1$，返回$y_1^* \leftarrow \{0, 1\}^n$。

3）最后，挑战者将(pk, y_b^*)发送给 A。

4. 猜测阶段：挑战者收到 A 对 b 的猜测 b′。

$Expt_A^2$：除了对v^*的定义外，本实验与 **$Expt_A^1$**相同。在本实验中，挑战者随机选取$v^* \leftarrow \{0, 1\}^{n+p}$并且解析$v^* = m^*||r^*$使得$m^*$为$v^*$的前 n 比特，$r^*$为$v^*$的后 p 比特。值得注意的是，$c^* = Com(m^*;r^*)$和$y^* = m^*$也相应地进行更新。本实验具体的描述如下：

1. 准备阶段：挑战者运行 $k \leftarrow CPRF.Gen(1^\lambda)$，设置私钥 $sk = k$。然后初始化一个空集合 $G = \emptyset$。

2. 询问阶段：挑战者回答 A 的以下询问：

- 求值询问：敌手 A 用定义域空间空间内的一个点 $x \in X$ 询问挑战者，挑战者首先计算 $v = F(k,x)$，然后解析 $v = m||r$ 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特，最后返回 $(y, \pi) = (m, r)$并更新集合 $G = G \cup \{x\}$。
- 受限密钥询问：敌手 A 选择一个受限集合 $S \in S_\lambda$询问挑战者，挑战者首先运行 $k_S \leftarrow CPRF.Cons(k, S)$，然后定义电路 $ConK[k_S]$（如图 2），并将其混淆后的版本设置为受限密钥 $sk_S \leftarrow O(\lambda, ConK_{k_S})$，最后返回 sk_S并更新集合 $G = G \cup \{x:S(x) = 1\}$。

3. 挑战和回复 pk 阶段：挑战者收到 A 的一个挑战询问$x^* \in X$，然后按照以下步骤运行：

1）挑战者首先随机选取$v^* \leftarrow \{0, 1\}^{n+p}$并解析$v^* = m^*||r^*$使得$m^*$为$v^*$的前 n 比特，$r^*$为$v^*$的后 p 比特，后直接生成一个承诺值$c^* = Com(m^*;r^*)$；之后，挑战者定义一个集合$S^* = X/\{x^*\}$并生成受限密钥$CPRF.Cons(k, S^*) \rightarrow k_{S^*}$。最后，挑战者定义电路 $PubK[x^*,c^*,k_{S^*}]$（如图 3）并将其混淆后的版本设置为公钥 $pk \leftarrow O(\lambda, PubK[x^*,c^*,k_{S^*}])$。

2） 挑战者先随机选取一个比特 $b \in \{0, 1\}$，若 $b = 0$，挑战者返回$y^* = m^*$；若 $b = 1$，返回$y_1^* \leftarrow \{0, 1\}^n$。

3）最后，挑战者将(pk, y_b^*)发送给 A。

4. 猜测阶段：挑战者收到 A 对 b 的猜测 b′。

$Expt_A^3$：除了对y_0^*的定义外，本实验与 **$Expt_A^2$**相同。本实验具体的描述如下：

1. 准备阶段：挑战者运行 $k \leftarrow CPRF.Gen(1^\lambda)$，设置私

钥 sk = k。然后初始化一个空集合 G = ∅。

2. 询问阶段：挑战者回答 A 的以下询问：

- 求值询问：敌手 A 用定义域空间空间内的一个点 x ∈ X 询问挑战者，挑战者首先计算 v = F(k,x)，然后解析 v = m||r 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特，最后返回 (y, π) = (m, r)并更新集合 G = G ∪ {x}。
- 受限密钥询问：敌手 A 选择一个受限集合 S ∈ S_λ询问挑战者，挑战者首先运行 $k_S \leftarrow$CPRF.Cons(k, S)，然后定义电路 ConK[k_S]（如图 2），并将其混淆后的版本设置为受限密钥 $sk_S \leftarrow O(\lambda, ConK_{k_S})$，最后返回 sk_S并更新集合 G = G ∪ {x:S(x) = 1}。

3. 挑战和回复 pk 阶段：挑战者收到 A 的一个挑战询问$x^* \in X$，然后按照以下步骤运行：

1) 挑战者首先随机选取$v^* \leftarrow \{0,1\}^{n+p}$并解析$v^* = m^*||r^*$使得$m^*$为$v^*$的前 n 比特，$r^*$为$v^*$的后 p 比特，后直接生成一个承诺值$c^* = Com(m^*;r^*)$；之后，挑战者定义一个集合$S^* = X/\{x^*\}$并生成受限密钥 $CPRF.Cons(k, S^*) \rightarrow k_{S^*}$。最后，挑战者定义电路 $PubK[x^*,c^*,k_{S^*}]$（如图 3）并将其混淆后的版本设置为公钥 $pk \leftarrow O(\lambda, PubK[x^*,c^*,k_{S^*}])$。

2) 挑战者先随机选取一个比特 b ∈ {0, 1}，若 b = 0，挑战者返回$y^* = m^*$；若 b = 1，返回$y_1^* \leftarrow \{0,1\}^n$。

3) 最后，挑战者将(pk, y_b^*)发送给 A。

4. 猜测阶段：挑战者收到 A 对 b 的猜测 b′。

我们接下来给出一些引理来证明每两个相邻实验的不可区分性，并表明在最后一个实验中，敌手攻击成功的优势是可忽略的，由此得出 $\mathbf{Expt}_A^0$中敌手攻击成功的优势也是可忽略的。

引理 1. 如果 F 满足 CPRF 的正确性，O 满足混淆器的不可区分性，那么有$|Adv_A^1 - Adv_A^0| \le negl(\lambda)$。

证明：若存在 PPT 的敌手 A 使得上述公式不成立，那么我们可以构造两个敌手(Samp,D)去攻击混淆器的不可区分性。除了 pk 的生成外，敌手与 $\mathbf{Expt}_A^0$中的挑战者行为一致。具体来讲，当 Samp 收到 A 的挑战 x^* 后，运行以下步骤：

1) 运行 $Samp(1^\lambda) \rightarrow (C_0, C_1, \sigma)$使得 $Pr\left[|C_0| = |C_1| \wedge \forall x, C_0(x) = C_1(x)\right] \ge 1 - negl(\lambda)$，其中$C_0$，$C_1$分别等于电路 PubK[k]和 $PubK[c^*,x^*,sk_{S^*}]$。

2) 随后，Samp 发送(C_0, C_1, σ)给挑战者，挑战者先随机抽取一个比特 t ∈ {0, 1}，然后计算$\hat{C}_t \leftarrow O(\lambda, C_t)$并将其返回给 D。

3) D 设置公钥 $pk = \hat{C}_t$并将其发送给 A。

D 之后的运行步骤仍等于 $\mathbf{Expt}_A^0$中的挑战者，唯一的区别是，在 D 收到 A 的猜测 b′后，其输出$t' = 1$当且仅当$b' = b$，否则$t' = 0$。

我们观察到，当 t = 0 时，A 的视图等价于 $\mathbf{Expt}_A^0$；当 t = 1 时，其视图相当于 $\mathbf{Expt}_A^1$。那么，

$$\begin{aligned}
&|Pr\left[D\left(\hat{\sigma}, C_o\right) = 1\right] - Pr\left[D\left(\hat{\sigma}, C_1\right) = 1\right]| \\
&= \left|Pr\left[Expt_A^0 = 1\right] - Pr\left[Expt_A^1 = 1\right]\right| \\
&= |(Pr\left[Expt_A^0 = 1\right] - \frac{1}{2}) - (Pr\left[Expt_A^1 = 1\right] - \frac{1}{2})| \\
&\ge |Adv_A^0 - Adv_A^1|
\end{aligned}$$

由于 CPRF 的正确性，即，任意的 $x \neq x^*$，都有 $CPRF.CEval\left(k_{S^*}, x\right) = F(k, x)$，因此电路$C_0$,$C_1$是等价的。由混淆器的不可区分性得知，D 区分$\hat{C}_t$的优势是可忽略的，因此有$\left|Adv_A^1 - Adv_A^0\right| \le negl(\lambda)$。

引理 2. 如果 F 满足 CPRF 的动态安全性，则有$\left|Adv_A^1 - Adv_A^2\right| \le negl(\lambda)$。

证明：如果存在一个 PPT 的敌手 A 使得上述等式不成立，那么我们可以构造一个敌手 D 去攻击 CPRF 的动态安全性！D 与实验 $\mathbf{Expt}_A^1$中的挑战者运行一致，除了 D 可以通过与 CPRF 挑战者交互的方式来回答 A 的所有询问，包括求值询问、受限密钥询问以及一个单独的挑战询问。这主要是因为 D 不拥有 CPRF 的私钥 k。D 按照如下方式运行：

1) D 初始化一个空集合 G = ∅，然后回答 A 的以下询问：

- 求值询问：当敌手 A 用定义域空间空间内的一个点 x ∈ X 询问 D 时，D 直接将 x 发送给 CPRF 挑战者。挑战者运行 $k \leftarrow CPRF.Gen(1^\lambda)$并计算 v = F(k, x)，随后将 v 返回给 D。D 解析 v = m||r 使得 m 为 v 的前 n 比特，r 为 v 的后 p 比特，然后将(y, π) = (m, r)发送给 A 并更新集合 G = G ∪ {x}。
- 受限密钥询问：当 A 选择一个受限集合 S ∈ S_λ询问 D，D 直接将 S 发送给挑战者并得到挑战者的回复$k_S \leftarrow CPRF.Cons(k, S)$。然后 D 生成一个电路 ConK[$k_S$](如图 2)以及相应的受限密钥 $sk_S \leftarrow O(\lambda, ConK[k_S])$。最后 D 将 sk_S发送给 A，并更新集合 D = D ∪ {x:S(x) = 1}。

2) 当 D 收到 A 的一个挑战询问$x^* \in X$，D 按照以下步骤运行：

a) D 发送$S^* = X/\{x^*\}$给挑战者，挑战者运行

$k_{S'}\leftarrow$CPRF.Cons(k, S^*)并将$k_{S'}$返回给 D。

b) D 发送x^*给挑战者，挑战者计算$v_0^* = F'(k, x^*)$，$v_1^*\leftarrow\{0,1\}^{n+p}$，然后抽取一个比特 $t\leftarrow\{0,1\}$并将v_t^*发送给 D。

c) D 解析$v_t^* = m_t^* || r_t^*$使得m_t^*为v_t^*的前 n 比特，r_t^*为v_t^*的后 p 比特，并设置$c^* = \mathrm{Com}(m_t^*; r_t^*)$。然后 D 生成一个电路 PubK[$x^*$,$c^*$,$k_{S'}$](如图 3)并且设置公钥 pk←O(λ,PubK[x^*,c^*,$k_{S'}$])。

d) 最后 D 随机抽取一个比特 b 并且设置$y_b^* = m_t^*$，然后将(y_b^*, pk)发送给 A。

3) 当 D 收到 A 对于 b 的猜测 b′，判定如果$b' = b$，D 则输出 1，否则 D 输出 0。

我们观察到，当 t = 0，A 的视图等价于实验 $\mathbf{Expt}_A^1$，此时只有当 A 的猜测$b' \neq b$，D 才会输出$t' = t = 0$。当 t = 1，A 的视图等价于实验 $\mathbf{Expt}_A^2$，此时只有当 A 的猜测$b' = b$，D 才会输出$t' = t = 1$。D 攻击成功的优势计算如下：

$$\begin{aligned}
&\mathrm{Adv}_D = \left|\Pr\left[\mathrm{Expt}_D^{CPRF}:t'=t\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[t' = 0 \wedge t = 0\right] + \Pr\left[t'=1 \wedge t=1\right] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\cdot\Pr\left[t'=0 \middle| t=0\right] + \frac{1}{2}\cdot\Pr\left[t'=1 \middle| t=1\right] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\cdot\left(1-\Pr\left[\mathrm{Expt}_A^1:b'=b\right]\right) + \frac{1}{2}\cdot\Pr\left[\mathrm{Expt}_A^2:b'=b\right] - \frac{1}{2}\right| \\
&= \frac{1}{2}\cdot\left|\Pr\left[\mathrm{Expt}_A^2:b'=b\right] - \Pr\left[\mathrm{Expt}_A^1:b'=b\right]\right| \\
&\geq \frac{1}{2}\cdot\left|\mathrm{Adv}_A^2 - \mathrm{Adv}_A^1\right|
\end{aligned}$$

由 CPRF 的动态伪随机性可知敌手 D 在 CPRF 的安全实验中成功的优势是可忽略的，由此可以得出$\left|\mathrm{Adv}_A^2 - \mathrm{Adv}_A^1\right|$成功优势也是可忽略的，因此引理 2 得证！

引理 3. 如果 Com 满足承诺方案的计算隐藏性，则有$\left|\mathrm{Adv}_A^2 - \mathrm{Adv}_A^3\right| \leq \mathrm{negl}(\lambda)$。

证明. 如果这里存在一个敌手 A 使得上述不等式不成立，那么我们可以构造一个敌手 D 去攻击承诺方案的隐藏性。具体来说：

1) 对于敌手 A 的每次求值询问或者受限密钥询问，D 和 Expt_A^2中的挑战者表现一致。

2) 对于 A 的某个挑战询问，D

a) 随机抽取一个$v^*\leftarrow\{0,1\}^{n+p}$并解析$v^* = m^* || r^*$使得$m^*$为$v^*$的前 n 比特，$r^*$为$v^*$的后 p 比特，设置$m_0 = m^*$。并随机抽取$m_1\leftarrow\{0,1\}^n$。

b) D 发送(m_0, m_1)给承诺方案的挑战者。挑战者随机抽取一个比特的 $t\leftarrow\{0,1\}$以及一个随机数$r\leftarrow\{0,1\}^p$并且返回承诺值$c_t^* = \mathrm{Com}(m_t; r)$。

c) D 利用c_t^*生成一个电路 PubK[x^*,c_t^*, $k_{S'}$]（如图 3），并生成 pk←O(λ,PubK[x^*,c^*,$k_{S'}$])。

d) 最后 D 随机抽取一个比特 b 并且设置$y_b^* = m_t$，然后将(y_b^*, pk)发送给 A。D 收到 A 对 b 的猜测 b′后，输出$t' = b'$。

我们观察到，当 b = t 时，A 的视图等价于实验 $\mathbf{Expt}_A^2$，此时 D 成功猜测$t' = t$的概率等价于 A 成功猜测$b' = b$的概率。当 b ≠ t 时，A 的视图等价于 $\mathbf{Expt}_A^3$，此时 D 成功猜测$t' = t$的概率等价于 A 成功猜测$b' \neq b$的概率。因此，有以下不等式：

$$\begin{aligned}
&\left|\Pr\left[t'=t:D(m_0,m_1,c_t^*)\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[t = b \wedge t' = t\right] + \Pr\left[t\neq b \wedge t'=t\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[t=b\right]\cdot\Pr\left[t'=t \middle| t=b\right] + \Pr\left[t\neq b\right]\cdot\Pr\left[t'=t \middle| t\neq b\right] - \frac{1}{2}\right| \\
&= \left|\frac{1}{2}\cdot\Pr\left[\mathrm{Expt}_A^2:b'=b\right] + \frac{1}{2}\cdot\left(1-\Pr\left[\mathrm{Expt}_A^3:b'=b\right]\right) - \frac{1}{2}\right| \\
&= \frac{1}{2}\cdot\left|\Pr\left[\mathrm{Expt}_A^2:b'=b\right] - \Pr\left[\mathrm{Expt}_A^3:b'=b\right]\right| \\
&\geq \frac{1}{2}\cdot\left|\mathrm{Adv}_A^2 - \mathrm{Adv}_A^3\right|
\end{aligned}$$

由于承诺方案的隐藏性，D 成功的优势是可忽略的，因此有$\left|\mathrm{Adv}_A^2 - \mathrm{Adv}_A^3\right| \leq \mathrm{negl}(\lambda)$。

引理 4. 在实验 Expt_A^3中，敌手成功的优势为 0，即 $\mathrm{Adv}_A^3 = 0$。

证明. 观察发现，无论 b 是什么，挑战者总是输出一个随机值$y^*\leftarrow\{0,1\}^n$，因此 A 成功猜测$b' = b$的概率是$\frac{1}{2}$，即$\mathrm{Adv}_A^3(\lambda) = \left|\Pr\left[\mathrm{Expt}_A^3(\lambda)=1\right] - \frac{1}{2}\right| = 0$。

结合以上所有引理得 $\mathrm{Adv}_A^0 \leq \mathrm{negl}(\lambda)$，定理 3 得证。

5 总结

PRF 是现代密码学的基本原语之一，被广泛应用于许多密码方案或密码协议的构造中。根据不同的应用需求，学者们在标准 PRF 的基础上增加了许多其他的性质，从而得到不同的 PRF 变体，对相关变体的研究是理论密码学中一个重要的研究方向！本文所研究的 CPRF 和 CVRF 即为 PRF 的变体之二！

CPRF 是在 PRF 的基础上增加了受限委托的能力，即 PRF 私钥持有者可对 PRF 定义域空间内的任意子集生成一个受限密钥，该密钥可被用来计算子集内的任意输入点处的 PRF 输出。CVRF 在 PRF 的基础上同时增加受限委托性质和可验证性，也被看作在 CPRF 的基础上增加了可验证性。从 CPRF 的角度来说，CVRF 要求，对任意的 CPRF 输出，无论其是由受限密钥计算还是私钥计算的，都必须同时伴随一个对输出的证明；任意拥有证明以及 CPRF 一些公钥信息的一方，都可以对输出的正确性进行验证。

尽管目前 CPRF 和 CVRF 的相关研究已经取得了一定的进展，但仍存在一些问题，比如 CPRF 相关原语种类繁多，命名交错，使用混乱；目前大多数 CVRF 构造要么只实现了较弱的安全性，要么在实现稍强安全性的同时存在一定的归约损失等。本文在一定程度上解决了以上的两个问题。

首先，我们对 CPRF 的定义进行了清晰的梳理，并着重回答了相关的歧义点问题。具体来说，表明 CPRF 与 F-PRF 并不等价，CPRF 也无法代替 F-PRF；证明 CPRF 单挑战安全性与多挑战安全性等价并给出了完整的安全性归约证明过程；说明 CPRF 正确性和其他额外属性的定义等。在深刻理解 CPRF 定义的基础上，我们对 CPRF 的可验证性，即 CVRF 进行研究。本文依赖于不可区分混淆、承诺方案和 CPRF，给出了 CVRF 的一般构造，该构造满足 CVRF 的半动态安全性，且具备两个额外的优势：支持任意有效可表达的集合，安全归约证明是紧致的，即归约损失为常数。

除此之外，我们提出开放性问题：（1）当 F-PRF 中函数 g 的值域无法有效判定时，如何给出通用的（或者具体的）安全性归约证明技术；（2）如何给出动态安全的、紧归约的、针对任意受限集合的 CVRF 构造。

致　谢　感谢李红达老师对本文 CPRF 定义的讨论等内容的指导，提出了许多建设性意见，这对本文内容的完善起到了至关重要的作用。感谢孟宪宁师姐对本文写作方面，包括规范用词、合理安排逻辑框架等工作的建议和帮助。感谢梁蓓老师的课题资助。

参考文献

[1] Goldreich O, Goldwasser S, Micali S. How to construct random functions[J]. *Journal of the Acm (JACM)*,1986, 33(4): 792-807.

[2] Bogdanov A, Rosen A. Pseudorandom functions: Three decades later[M]. Tutorials on the Foundations of Cryptography, 2017: 79-158.

[3] Micali S, Rabin M, Vadhan S. Verifiable random functions[C]. *40th annual symposium on foundations of computer science (cat. No. 99CB37039). IEEE*, 1999: 120-130.

[4] Naor M, Pinkas B, Reingold O. Distributed Pseudorandom Functions and KDCs[J]. *International Conference on the Theory and Application of Cryptographic Techniques*, 1999, 327-346.

[5] Boneh D, Kim S, Wu D J. Constrained keys for invertible pseudorandom functions[C]. *Theory of Cryptography Conference*, 2017: 237-263.

[6] M. Naor, O. Reingold. Number-theoretic constructions of efficient pseudo-random functions[J]. *Journal of the ACM (JACM)*, 2004, 51(2): 231–262.

[7] Freedman M J, Ishai Y, Pinkas B, et al. Keyword Search and Oblivious Pseudorandom Functions[C]. *Second Theory of Cryptography*, 2005: 303–324.

[8] Jarecki S, Liu X. Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection[C]. *Theory of Cryptography Conference: 6th Theory of Cryptography Conference*, 2009: 577-594.

[9] Boneh D, Waters B. Constrained pseudorandom functions and their applications[C]. *19th International Conference on the Theory and Application of Cryptology and Information Security*, 2013: 280-300.

[10] Boyle E, Goldwasser S, Ivan I. Functional signatures and pseudorandom functions[C]. *International workshop on public key cryptography*, 2014: 501-519.

[11] Kiayias A, Papadopoulos S, Triandopoulos N, et al. Delegatable pseudorandom functions and applications[C]. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, 2013: 669-684.

[12] Sahai A, Waters B. How to use indistinguishability obfuscation: deniable encryption, and more[C]. *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014: 475-484.

[13] Fuchsbauer G. Constrained verifiable random functions[C]. *Security and Cryptography for Networks*, 2014: 95-114.

[14] Gilad Y, Hemo R, Micali S, et al. Algorand: Scaling byzantine agreements for cryptocurrencies[C]. *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017: 51-68.

[15] Liu M, Zhang P, Wu Q. A Novel Construction of Constrained Verifiable Random Functions[J]. *Security and Communication Networks*, 2019: 1-15.

[16] Chandran N, Raghuraman S, Vinayagamurthy D. Constrained Pseudorandom Functions: Verifiable and Delegatable[J]. *Iacr Cryptology Eprint Archive*, 2014: 522.

[17] Datta P. Constrained (Verifiable) Pseudorandom Function from Functional Encryption[C]. *Information Security Practice and Experience*, 2018: 141-159.

[18] Datta P. Constrained Pseudorandom Functions from Functional Encryption[J]. *Theoretical Computer Science,* 2020, 809: 137-170.

[19] Datta P, Dutta R, Mukhopadhyay S. Constrained Pseudorandom Functions for Unconstrained Inputs Revisited: Achieving Verifiability and Key Delegation[C]. *Public-Key*

Cryptography, 2017: 463-493.

[20] Datta P, Dutta R, Mukhopadhyay, S. Constrained pseudorandom functions for turing machines revisited: How to achieve verifiability and key delegation[J]. *Algorithmica*, 2019: 81: 3245-3390.

[21] Liang B, Li H, Chang J. Constrained verifiable random functions from indistinguishability obfuscation[C]. *International Conference on Provable Security*, 2015: 43-60.

[22] Reyneri J, Karnin E. Coin flipping by telephone (Corresp.)[J]. *IEEE Transactions on Information Theory*, 1984, 30(5): 775-776.

[23] Bitansky N. Verifiable random functions from non-interactive witness indistinguishable proofs[J]. *Journal of Cryptology*, 2013, 33(3): 459-493.

[24] Davidson A, Katsumata S, Nishimaki R, et al. Adaptively secure constrained pseudorandom functions in the standard model[C]. *Annual International Cryptology Conference*, 2020: 559-589.

[25] Attrapadung N, Matsuda T, Nishimaki R, et al. Constrained PRFs for NC1 in Traditional Groups (from CRYPTO 2018)[J]. *IEICE Technical Report*, 2018, 118(212): 61-61.

[26] Davidson A, Katsumata S, Nishimaki R, et al. Constrained PRFs for Bit-fixing from OWFs with Constant Collusion Resistance[J]. *IACR Cryptol. ePrint Arch*, 2018: 982.

[27] Hofheinz D. Fully secure constrained pseudorandom functions using random oracles[J]. *Cryptology ePrint Archive*, 2014: 372.

[28] Hofheinz D, Kamath A, Koppula V, et al. Adaptively secure constrained pseudorandom functions[C]. *International Conference on Financial Cryptography and Data Security*, 2019: 357-376.

[29] Fuchsbauer G, Konstantinov M, Pietrzak K, et al. Adaptive security of constrained PRFs[C]. *20th International Conference on the Theory and Application of Cryptology and Information Security*, 2014: 82-101.

[30] Abusalah H, Fuchsbauer G, Pietrzak K. Constrained PRFs for unbounded inputs[C]. *Cryptographers' Track at the RSA Conference*, 2016: 413-428.

[31] Abusalah H, Fuchsbauer G. Constrained PRFs for unbounded inputs with short keys[C]. *Applied Cryptography and Network Security: 14th International Conference*, 2016: 445-463.

[32] Chandran N, Raghuraman S, Vinayagamurthy D. Reducing Depth in Constrained PRFs: From Bit-Fixing to NC1[C]. *19th IACR International Conference on Practice and Theory in Public-Key Cryptography*, 2016: 359-385.

[33] Kalai Y, Reyzin L. Private Constrained PRFs (and More) from LWE[C]. *Theory of Cryptography - 15th International Conference,* 2017: 264-302.

[34] Canetti R, Chen Y. Constraint-hiding constrained PRFs for NC from LWE[C]. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2017: 446-476.

[35] Boneh D, Lewi K, Wu D J. Constraining pseudorandom functions privately[C]. *IACR International Workshop on Public Key Cryptography*, 2017: 494-524.

[36] Peikert C, Shiehian S. Privately constraining and programming PRFs, the LWE way[C]. *IACR International Workshop on Public Key Cryptography*, 2018: 675-701.

[37] Attrapadung N, Matsuda T, Nishimaki R, et al. Adaptively single-key secure constrained PRFs for NC1[C]. *IACR International Workshop on Public Key Cryptography*, 2019: 223-253.

[38] Zan Y, Li H, Meng X, et al. Generic Construction of (Hierarchical) Functional Pseudorandom Functions[C]. *Proceedings of the 2022 3rd International Conference on Control, Robotics and Intelligent System*, 2022: 171-175.

[39] Garg S, Gentry C, Halevi S, et al. Candidate indistinguishability obfuscation and functional encryption for all circuits[J]. *SIAM Journal on Computing*, 2016, 45(3): 882-929.

訾瑶于 2017 年在河南农业大学电子信息科学与技术（信息与安全）专业获得学士学位。现在中国科学院大学网络空间安全专业攻读博士学位。研究领域为理论密码学、可证明安全理论。研究兴趣包括：伪随机函数。Email：zanyao@iie.ac.cn

梁蓓于 2016 年在中国科学院大学信息安全专业获得博士学位。现任北京雁栖湖应用数学研究院助理教授。研究领域为公钥密码学，可证明安全密码体制。 研究兴趣包括：安全多方计算。Email: lbei@bimsa.cn