Injective TDF to CCA

很久没写了小短文了，今天介绍一个理论的构造，相信从**带有性质**的TDF构造CCA安全的PKE方案大家都很熟悉了，比如：

- "Doubly Enhanced" Permutation的TDF可以蕴涵NIZK，结合Naor-Yung的两把钥匙的构造思想（需要强化NIZK为one-time simulation的，例如可以参考DDN构造），从而可以构造CCA安全的PKE方案
- 著名的 Lossy Trapdoor Function（ABO版本）可以很简洁地构造CCA安全的PKE方案
- Correlated Product Secure的TDF也可以构造CCA安全的PKE方案
- Adaptive Trapdoor Function也可以用来构造CCA安全的PKE方案（$\text{L-TDF} \Rightarrow \text{CP-TDF} \Rightarrow \text{A-TDF}$）

那么一个问题在于，我们能否从更弱的TDF，不具有那么多结构的TDF出发，构造CCA安全的PKE呢？

今天介绍的CRYPTO 2020年获得Best Paper Award的论文：**Chosen Ciphertext Security from Injective Trapdoor Functions**，该论文由 Hohenberger, Koppula, Waters 完成，其说明了从**仅具有单射性质**的TDF出发，也可以黑盒地构造CCA安全的PKE方案。

> *从具有单射性质的TDF出发，利用GL引理可以很方便且黑盒地构造一比特的伪随机数，从而可以直接构造CPA安全的PKE.*
>
> *笔者注：GL引理在OWF下也成立*

本篇论文主要的三个组件：

- 可恢复随机数的PKE（依赖于Injective TDF的Injectivity和Trapdoor）
- 针对固定长度集合的Commitment（PRG就可以构造，也就是OWF即可）
- 强一次抗伪造签名

1 三个组件

可恢复随机数的加密：

- 解密算法$\mathsf{Dec}(sk, c)$不仅可以恢复出消息m，**还可以恢复出**加密消息c使用的随机数r（感觉较难实现：例如ElGamal不具有这样的性质）
- $\mathsf{Recover}(c, r)$，若r是c加密时候的随机数则可恢复出消息m（感觉容易实现：$\mathsf{Enc}(pk, r)||r \oplus m$）

CPA安全的PKE的经典构造$\big(f(x), r, (x \oplus r) \oplus m\big)$显然是**同时**满足上述的性质（其先恢复$r$）

可进一步ℓ_{msg}比特长度:

- KeyGen: $t \leftarrow \{0,1\}^{\ell_{inp}}, pk = (tdf.pk, t), sk = (tdf.sk, t)$
- Enc($pk, \mathbf{m} = (m_1, \cdots, m_{\ell_{msg}})$):
  - $r_i \leftarrow \{0,1\}^{\ell_{inp}}$
  - $ct_{1,i} = r_i \cdot t + m_i, ct_{2,i} = tdf.eval(tdf.pk, r_i)$
- Dec: 给sk显然会恢复r_i（加密用随机数），然后再解密得到m
- Recover: 有了(r, c)，显然可以算出m

ElGamal类则是直接在群上进行运算，不会恢复出$\mathbb{Z}_p$上的随机数，其Dec算法不具有如此恢复随机数的性质。

带标签的集合承诺:

- 给一个B大小的集合 $S \in [N]$ 产生承诺（在某一个标签下），包括两个部分:
  - 整体的承诺 com
  - 每一个元素的承诺 σ_i
  - 验证性质（验证序号i是否在S中）：Verify($pp, com, i, \sigma_i, tag$)
- 安全性质（在某个标签下）：
  - Setup的不可区分性：敌手任意选择S，为S用正常方式产生的，和另一种 AltSetup 方式产生的承诺不可区分（该setup方式仅用于安全证明——为了去除 S 的**特殊性**）
  - Soundness：敌手看到了一个某个tag在AltSetup下产生的承诺，也无法在同一个pp下的**另一个标签**tag^*下产生为一个大小**超过**B的集合S产生“合法”承诺

强一次抗伪造签名:

- 利用Lamport构造

2 组件的具体性质及其构造

可恢复随机数的加密的语义:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$: The setup algorithm takes as input the security parameter λ and outputs a public key pk and secret key sk.

$\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$: The encryption algorithm is randomized; it takes as input a public key pk and a message m, uses ℓ_{rnd} bits of randomness and outputs a ciphertext ct. We will sometimes write $\mathsf{Enc}(\mathsf{pk}, m; r)$, which runs $\mathsf{Enc}(\mathsf{pk}, m)$ using r as the randomness.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to z \in (\{0,1\}^{\ell_{\mathrm{msg}}} \times \{0,1\}^{\ell_{\mathrm{rnd}}}) \cup \{\perp\}$: The decryption algorithm takes as input a secret key sk and a ciphertext ct, and either outputs $z = \perp$ or $z = (m, r)$ where $m \in \{0,1\}^{\ell_{\mathrm{msg}}}$, $r \in \{0,1\}^{\ell_{\mathrm{rnd}}}$.

$\mathsf{Recover}(\mathsf{pk}, \mathsf{ct}, r) \to z \in \{0,1\}^{\ell_{\mathrm{msg}}} \cup \{\perp\}$: The recovery algorithm takes as input a public key pk, a ciphertext ct and string $r \in \{0,1\}^{\ell_{\mathrm{rnd}}}$. It either outputs $\perp$ or a message $m \in \{0,1\}^{\ell_{\mathrm{msg}}}$.

- 其构造就采用在第 1 小节当中描述的思路进行
- 其安全性是CPA安全

带标签的集合承诺：

$\mathsf{Setup}(1^\lambda, 1^N, 1^B, 1^t) \to \mathsf{pp}$: The setup algorithm takes as input the security parameter λ, the universe size N, bound B on committed sets and tag length t, and outputs public parameters pp.

$\mathsf{Commit}(\mathsf{pp}, S \subseteq [N], \mathsf{tg} \in \{0,1\}^t) \to (\mathsf{com}, (\sigma_i)_{i \in S})$: The commit algorithm is randomized; it takes as input the public parameters pp, set S of size B and string tg, and outputs a commitment com together with 'proofs' σ_i for each $i \in S$. [4]

$\mathsf{Verify}(\mathsf{pp}, \mathsf{com}, i \in [N], \sigma_i, \mathsf{tg} \in \{0,1\}^t) \to \{0,1\}$: The verification algorithm takes as input the public parameters, an index i, a proof σ_i, and tg. It outputs $0/1$.

$\mathsf{AltSetup}(1^\lambda, 1^N, 1^B, 1^t, \mathsf{tg}) \to (\mathsf{pp}, \mathsf{com}, (\sigma_i)_{i \in [N]})$: The scheme also has an 'alternate setup' which is used in the proof. It takes the same inputs as Setup together with a special tag tg, and outputs public parameters pp, commitment com together with proofs σ_i for all $i \in [N]$.

These algorithms must satisfy the following perfect correctness requirements:

Correctness of Setup and Commit: For all λ, N, $B \leq N$, t, $\mathsf{tg} \in \{0,1\}^t$ and set $S \subseteq [N]$ of size B, if $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^N, 1^B, 1^t)$ and $(\mathsf{com}, (\sigma_i)_{i \in S}) \leftarrow \mathsf{Commit}(\mathsf{pp}, S, \mathsf{tg})$, then for all $i \in S$, $\mathsf{Verify}(\mathsf{pp}, \mathsf{com}, i, \sigma_i, \mathsf{tg}) = 1$.

Correctness of $\mathsf{AltSetup}$: For all λ, N, $B \leq N$, t, $\mathsf{tg} \in \{0,1\}^t$, if $(\mathsf{pp}, \mathsf{com}, (\sigma_i)_{i \in [N]}) \leftarrow \mathsf{AltSetup}(1^\lambda, 1^N, 1^B, 1^t, \mathsf{tg})$, then for all $i \in [N]$, $\mathsf{Verify}(\mathsf{pp}, \mathsf{com}, i, \sigma_i, \mathsf{tg}) = 1$.

- 安全性质
  - Setup的**可替换性**，取消 S 的特殊性

    Definition 4.1. A tagged set commitment scheme $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify}, \mathsf{AltSetup})$ satisfies indistinguishability of setup if for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $|\Pr[1 \leftarrow \mathsf{Expt\text{-}Ind\text{-}Setup}_{\mathcal{A}}(\lambda)] - 1/2| \leq \mathsf{negl}(\lambda)$, where $\mathsf{Expt\text{-}Ind\text{-}Setup}_{\mathcal{A}}$ is defined in Figure 1.

    $\mathsf{Expt\text{-}Ind\text{-}Setup}_{\mathcal{A}}(\lambda)$

    1. Adversary $\mathcal{A}$ receives input 1^λ and sends $1^N, 1^B, 1^t, \mathsf{tg}, S$ such that $B \leq N$, $\mathsf{tg} \in \{0,1\}^t$ and $|S| = B$.
    2. Challenger chooses $b \leftarrow \{0,1\}$. It computes $\mathsf{pp}^0 \leftarrow \mathsf{Setup}(1^\lambda, 1^N, 1^B, 1^t)$ and $(\mathsf{com}^0, (\sigma_i^0)_{i \in S}) \leftarrow \mathsf{Commit}(\mathsf{pp}, S, \mathsf{tg})$, and $(\mathsf{pp}^1, \mathsf{com}^1, (\sigma_i^1)_{i \in [N]}) \leftarrow \mathsf{AltSetup}(1^\lambda, 1^N, 1^B, 1^t, \mathsf{tg})$. It sends $(\mathsf{pp}^b, \mathsf{com}^b, (\sigma_i^b)_{i \in S})$ to $\mathcal{A}$.
    3. $\mathcal{A}$ outputs its guess b'. The experiment outputs 1 iff $b = b'$.

    Figure 1: Experiment for Indistinguishability of Setup

- **集合承诺的大小绑定性**：pp确定下仅能生成B个合法的承诺（不能在tag^*下超出B个承诺纵使在知道tag下B个合法承诺）

Definition 4.2. A tagged set commitment scheme Com = (Setup, Commit, Verify, AltSetup) satisfies soundness security if for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $\Pr[1 \leftarrow \mathsf{Expt\text{-}Sound}_{\mathcal{A}}(\lambda)] \le \mathsf{negl}(\lambda)$, where $\mathsf{Expt\text{-}Sound}_{\mathcal{A}}$ is defined in Figure 2.

$\mathsf{Expt\text{-}Sound}_{\mathcal{A}}(\lambda)$

1. Adversary $\mathcal{A}$ receives input 1^λ, sends $1^N, 1^B, 1^t, \mathsf{tg}$ such that $B \le N$, $\mathsf{tg} \in \{0,1\}^t$.
2. Challenger computes $(\mathsf{pp}, \mathsf{com}, (\sigma_i)_{i\in[N]}) \leftarrow \mathsf{AltSetup}(1^\lambda, 1^N, 1^B, 1^t, \mathsf{tg})$, sends $(\mathsf{pp}, \mathsf{com}, (\sigma_i)_{i\in[N]})$ to $\mathcal{A}$.
3. $\mathcal{A}$ outputs $\mathsf{tg}' \ne \mathsf{tg}$, set $S \subseteq [N]$ of size greater than B, commitment com' and proofs $(\sigma_i')_{i\in S}$. The experiment outputs 1 iff for all $i \in S$, $\mathsf{Verify}(\mathsf{pp}, \mathsf{com}', i, \sigma_i', \mathsf{tg}') = 1$.

Figure 2: Experiment for Soundness Security

构造思路：利用PRG + 单射编码函数$\Gamma : \{0,1\}^t \to \mathbb{F}_{2^\ell}$

- Setup: 产生N对$A_i, D_i \in \mathbb{F}_{2^\ell}$作为pp
- Commit(pp, $S \subset [N], tag$): 为每一个$i \in S$分配随机数$\sigma_i = s_i$，构造com为一个$B-1$次$\mathbb{F}_{2^\ell}$上的多项式，其中$p(i) = \mathsf{PRG}(s_i, 1^\ell) + A_i + D_i \cdot \Gamma(tag)$
- Verify(pp, com, i, σ_i, com, tag): 验证是否有$p(i) = \mathsf{PRG}(s_i, 1^\ell) + A_i + D_i \cdot \Gamma(tag)$
- Altsetup: 由于此时没有固定A_i, D_i，随机产生$B-1$次$\mathbb{F}_{2^\ell}$上的多项式$p(x)$作为com，随机采样N个D_i，定义$A_i := p(i) - \mathsf{PRG}(s_i, 1^\ell) - D_i \cdot \Gamma(tag)$

Setup 可以替换为 Altsetup:

- S当中分布相同：利用$p(x)$的随机性
- $[N]\backslash S$当中分布近似相同：利用PRG的伪随机性（在不暴露s_i的情况下）

大小绑定性：

- 注意到pp固定下来，且$\Gamma(tag^*) \ne \Gamma(tag)$，从而直觉上无法照搬Altsetup在$tag$下给出的$s_i$（或者说最多照搬$B$个$s_i$），到了$B+1$个的时候由于$p(x)$已经被完整地确定下来，必须要计算PRG的逆函数，直觉上这是计算困难的，更进一步如果我们将其与lossy trapdoor中的"信息丢失"这一概念相联系，那么**这是统计意义上困难的**（定义域指数级别地小于值域）
- 从统计上论证，我们需要证明这样的式子：

$$\rho(\lambda) = \Pr\left[\begin{array}{c} \exists p' \in \mathbb{F}_{2^\ell}[x]^{B-1}, S \subseteq [N], |S| = B+1, (s_i')_i, \mathsf{tg} \ne \mathsf{tg}' \text{ such that} \\ \forall i \in S, p(i) - \mathsf{PRG}(s_i, 1^\ell) - D_i \cdot \mathsf{emb}(\mathsf{tg}) = p'(i) - \mathsf{PRG}(s_i', 1^\ell) - D_i \cdot \mathsf{emb}(\mathsf{tg}') \end{array}\right]$$

这个式子的随即带建立在$p(x), s_i, D_i$的选择上，我们可以先固定$p(x), s_i$进行分析，一个信息论上的argument在于$p'(x), s_i', S$的熵在于$B \cdot \ell + (B+1)|s_i| + \log C_N^{B+1}$，而$D_i$的熵值为$(B+1)\cdot \ell$，只要让$\ell$充分大，则上述argument即可以存在，因为两个熵值不同的变量相等的概率非常小。

强一次抗伪造签名容易构造，略去

3 主方案构造与证明思路

主方案构造：利用 N 组（N为充分大的$\text{poly}(\lambda)$）可恢复随机数的CPA安全的PKE + 带标签的集合承诺 + 强一次抗伪造签名

- Setup: 产生 N 组密钥对 (pk_i, sk_i)，输入参数 B, N 产生带标签的集合承诺的pp
- Enc(pk, m): 产生签名公私钥(vk, ssk)，产生大小为 B 的随机 S 序号集合及其证明 $com, \{\sigma_i\}_{i\in S}$
  - 对 $i \in S$：Enc$(pk_i, 1|m|\sigma_i; r_i)$ 【保证$\bigoplus_{i\in S} r_i = 0$】
  - 对 $i \notin S$：Enc$(pk_i, 0)$ 【仅是为了方便添加在randomness中添加disinformation】
  - 做一次签名得到Σ
  - $(vk, \Sigma, com, \{ct_i\}_{i\in[N]})$
- Dec(sk, c):
  - 验证强一次抗伪造签名
  - 检测出S集合有且仅有B个，并提取出相应的加密信息
    - 依次用sk_i解密，得到(b, m_i, σ_i)和r_i
    - 首位是1，集合承诺通过则将(i, m_i, r_i)加入集合U
    - 验证U大小为B
  - 验证 $\bigoplus_U r_i = 0$
  - 重加密验证上述U集合应用r_i加密后仍然是ct_i

令挑战密文为$(vk^*, \Sigma^*, com^*, \{ct_i^*\}_{i\in[N]})$

证明思路：

- Game 1: 更换集合承诺的setup的setup方式
  - 从setup到Altsetup，为消除S的特殊化做准备
- Game 2: Dec Oracle拒绝vk^*的解密查询
- Game 3: 模拟Dec Oracle将挑战密文换为
  - 如下结构：
    - 对 $i \in S$：Enc$(pk_i, 1|m^*|\sigma_i^*; r_i^*)$ 【保证$\bigoplus_{i\in S} r_i^* = 0$】
    - 对 $i \notin S$：Enc$(pk_i, 1|m^*|\sigma_i^*)$ 【仅是为了方便添加在randomness中添加disinformation】
  - 需要分成N个过渡的Game

- Game $3.i \to$ Game $3.i+1$ $(i \in \{0, \cdots, N-1\})$ 将挑战密文中的第$i+1$个index下的ct_i替换为$\mathsf{Enc}(pk_i, 1|m^*|\sigma_i^*)$
    - 关键在于替换模拟的方式
    - Game $3.i+1$下没有sk_{i+1}，此时不急于解密ct_{i+1}，而是采用**等价的方式去模拟Dec Oracle**：
        - 验证强一次抗伪造签名
        - 检测出S集合有且仅有$B-1$或者B个，并提取出相应的加密信息（其他情况下拒绝）
            - 跳过sk_{i+1}依次用sk_j解密，得到(b, m_j, σ_j)和r_j
            - 首位是1，集合承诺通过则将(j, m_j, r_j)加入集合U
            - 验证U大小为B或者$B-1$
        - 若U大小为B，由于集合承诺的**大小绑定**性质不用进一步检验
        - 若U大小为$B-1$，利用$\mathbf{r}_i$恢复最后一个$\mathbf{r}_0$，从而可以利用Recover进行解密，从而进行剩余的检验
        - 其他大小输出 $\perp$
- Game 4: Game 3已经为挑战密文的randomness添加进足够的disinformation，本Game在于去除挑战密文中的加密随机数之间的**限制**（**CPA安全必须要概率加密**，去除随机数限制为了后续应用CPA安全性将m^*换为0）
    - 挑战密文变为如下结构
        - 对$i \in S$：$\mathsf{Enc}(pk_i, 1|m^*|\sigma_i^*; r_i^*)$　【$\bigoplus_{i \in S} r_i^*$没有限制，而是全随机】
        - 对$i \notin S$：$\mathsf{Enc}(pk_i, 1|m^*|\sigma_i^*)$　【从随机性的限制来说，S内和外都是自由的】
    - 统计论证，利用剩余哈希引理，考虑分布 $(\mathbf{r}, \oplus_{i:z_i=1}\mathbf{r}_i)$与$(\mathbf{r}, \mathbf{u})$，其中$\mathbf{r} = (\mathbf{r}_1, \cdots, \mathbf{r}_{N-1})$
- Game 5: 应用CPA安全性将m^*都替换0
    - 类似于Game 3的过渡Game

4 反思

我们离CPA黑盒地蕴涵CCA安全还有多远?

- 容易知道具有（近似）完美正确性的CPA方案蕴涵OWF
    - OWF蕴涵强一次抗伪造签名
    - OWF蕴涵PRG，从而蕴涵带标签的集合承诺
- 那么我们只差是否CPA安全的方案能否具有上述的可恢复随机数加密的良好性质?
    - 单独的Recover容易从CPA安全的PKE中黑盒地实现：$\mathsf{Enc}(pk, r)||r \oplus m$
- Dec具有随机数恢复的功能较难实现（随机数恢复是为了实现**重加密验证的功能**）

- 如何构造具有随机数恢复功能的PKE是一个问题!

High-Level Ideas:

- 其构造和证明思路在一定程度上仿造了Naor-Yung
  - 更精确地说，其为了证明 well-form，采用了“随机数绑定 + 可恢复加密随机数的PKE”的方式
  - 随机数绑定结合可恢复加密随机数PKE保证了只要可以**解密n**个并行加密中的$n-1$个，就能进行最后一个加密的解密（进行Dec Oracle的模拟）
    - 这种模拟方式还依赖于Dec算法的两个特点
      - 遇到错误不会终止，**只要找到B个合适的解密就能输出**
      - 集合承诺保证我们只需要检验到B个合适的就不用进一步检验
        - 其他check不通过的不用管
        - 不可能存在check通过的例子，由于集合承诺不能在新的tag下产生超过B个元素承诺σ_i
  - 这种$n-1$ out of n的方式可以分别地对不同的index进行替换，将0替换为m^*引入randomness的disinformation，利用LHL即可以去除随机数之间的依赖
  - 最后没有依赖的挑战消息可以很方便地利用CPA安全将m^*全部替换为0