解读RGB++ Layer四大特性：BTCFi与UTXO世界的枢纽

作者：Faust & 雾月，BTCEden

2024年7月，CKB官宣了RGB++ Layer的正式启动，标志着此前发布的RGB++协议彻底从理论落地为工程化产物，并将引入更具体、更实际的应用场景。凭借着在BTC与CKB、Cardano等泛UTXO公链之间构建BTCFi生态的愿景，RGB++ Layer很快成为了人们关注的焦点。

概括来说，RGB++ Layer以RGB++协议为基础，利用同构绑定和Leap技术，为RGB++原生资产或铭文/符文在BTC、CKB、Cardano等UTXO型公链之间提供“无需跨链桥”的全链交互体验；利用CKB图灵完备的智能合约环境，为比特币构建从资产发行到实现复杂DeFi功能的必要条件。

且由于RGB++ Layer背靠CKB完备的账户抽象生态，兼容比特币账户和钱包，可以为比特币用户创造良好的体验，为BTCFi的大规模采用铺平道路。

下文中，让我们深入理解RGB++ Layer的大致工作原理和特性，展望其为BTCFi生态带来的改变。由于其理论基础建立在RGB++协议之上，我们将先从协议本身开始讲起。



## RGB++协议：RGB++ Layer的理论基石

RGB++协议发布于今年1月，其核心理念是用CKB链上验证的形式，替代RGB协议的“客户端验证”，本质是把CKB当做去中心化的索引器，将数据存储、资产来源验证等任务交由CKB完成，由后者作为RGB协议的验证层和DA层，以解决RGB协议在UX上的弊病、不利于支持Defi的缺陷。

与“一次性封装”的概念相呼应，RGB++引入了同构绑定的概念，以CKB链上的拓展型UTXO——Cell作为铭文/符文类资产的数据载体，再令Cell与比特币/Cardano/Liquid链上的UTXO建立绑定关系，最终让RGB++资产继承比特币等UTXO公链的安全性，以防止发生双重支付。

这种“绑定XXX以继承XXX安全性”的思路，类似于现实中银行账户要绑定手机号和身份证。

举个例子，假设Alice要给BOB转去一些TEST代币，她可以生成一个声明，将存储TEST资产信息的Cell与Bob的比特币UTXO绑定起来。如果Bob打算再把TEST代币转给别人，绑定的比特币UTXO也要发生转移。

这样一来，承载RGB++资产数据的Cell，和比特币UTXO之间有1对1绑定的关系，只要比特币UTXO没有被双重消费，绑定的RGB++资产就不会被双花。



说到RGB++ Layer，它实际是对RGB++协议进行工程化落地的产物，其主打的两大特性，包括同构绑定和Leap无桥跨链，下面让我们深入了解下同构绑定和Leap的技术实现原理。

## 同构绑定与Leap：BTCFi的资产发行与无桥跨链层

为了真正理解同构绑定和Leap的思路，我们先简单说下CKB的Cell模型。

Cell实质是拓展型UTXO，有LockScript、TypeScript、Data等多个字段，LockScript的作用和比特币的锁定脚本类似，用于权限验证；TypeScript类似于智能合约代码，Data则用于存放资产数据。



如果你要在CKB链上发行RGB++资产，首先要创建一个Cell，并在相关字段里写好代币符号和合约代码，比如代币符号为TEST。之后你可以把这些Cell拆解，并分发给很多人，就和比特币UTXO的拆分和转移方式一样。

由于Cell与比特币UTXO在结构上相似，且CKB可以兼容比特币签名算法，用户可以用比特币钱包操纵CKB链上资产。假如你拥有某个Cell，你可以对锁定脚本进行设置，使解锁条件与比特币UTXO的解锁条件一致，这样就可以用比特币账户私钥操纵CKB链上的Cell。



上述特性在CKB、BTC和其他UTXO公链之间也可以实现，比如你也可以用Cardano账户改写CKB链上的资产数据，RGB++资产的控制权也可以从BTC账户转移到Cardano账户，而无需跨链桥。下面我们将对这个话题展开解释。

前面我们曾提到，RGB++资产需要绑定比特币、Cardano、Liquid等公链上的UTXO，类似于现实中银行账户要绑定手机号和身份证；其次，RGB++资产本身只是一堆数据，这些数据需要有数据库之类的存储媒介，CKB链上的Cell可以充当其数据库。

然后我们可以在权限验证这块做设置，允许人们用BTC、Cardano等不同公链的账户，去改写CKB链上的RGB++资产数据。这便是同构绑定的核心宗旨。

RGB++ Layer提出的“Leap”和无桥跨链，其实是基于同构绑定技术，对RGB++资产绑定的UTXO进行“换绑”，比如你的资产之前绑定了比特币UTXO，现在可以换绑到Cardano、Liquid、Fuel等链上的UTXO，这样就可以把资产控制权限从BTC账户转移到Cardano账户上。



从用户感知的角度看，这其实等价于资产跨链，CKB充当了类似于索引器和数据库的角色。但不同于传统的跨链方式，“Leap”只改变资产数据的使用权限，数据本身还是存储在CKB链上的，这种方式比Lock-Mint模式更简洁，也免去了对映射资产合约的依赖。

以上只是同构绑定和Leap的产品效果说明。下面让我们通过具体案例，来理解它们的技术实现思路。

## 同构绑定的实现方式

让我们来理解下同构绑定的技术实现方式。假设Alice有100枚TEST代币，数据存放在Cell#0 中，与比特币链上的UTXO#0 有绑定关系。

现在，Alice要把40枚TEST代币转给Bob。首先，她要把Cell#0拆分为两个新的Cell，其中Cell#1包含40枚TEST代币，转让给Bob；Cell#2包含60枚TEST，还是由Alice自己控制。

在这个过程中，Cell#0 绑定的BTC UTXO#0，也要拆分为UTXO#1 和UTXO#2，分别与Cell#1 和Cell#2绑定。当Alice把Cell#1转让给Bob时，可以一键操作把BTC UTXO#1也转让给Bob，在CKB和BTC链上实现同步交易。



我们可以在此深度理解下同构绑定。其实这个概念的核心意义在于，CKB的Cell、Cardano的eUTXO和BTC UTXO都是UTXO模型，且CKB兼容比特币/Cardano签名算法，后两条链上发生的UTXO的分解和转移，也可以1:1同步给CKB链上的Cell。

这样一来，当我们对绑定着RGB++资产的BTC UTXO进行操作时，可以把操作结果同步给CKB链上的Cell，就好像实体和影子的关系一样。另外我们也要注意，RGB++资产关联了BTC UTXO和CKB Cell这两个实体，两者都是RGB++资产的组成部分，缺一不可。



如果我们考察上面提到的Alice给Bob转账的案例，其大致流程为：

1. Alice在本地构造一笔CKB交易数据（先不上链），这笔交易指明将记录资产数据的Cell#0销毁，生成Cell#1送给Bob，Cell#2留给自己；

2. Alice在本地生成一个声明，把Cell#1绑定到BTC UTXO#1，把Cell#2绑定到BTC UTXO#2，并把Cell#1和BTC UTXO#1都送给Bob；

3.之后，Alice在本地生成一个Commitment（类似于hash），对应的原始内容包含第2步中的声明+第1步中生成CKB交易数据。Commitment的数据之后要被记录到比特币链上；

4.Alice在比特币链上发起交易，把UTXO#0销毁，生成UTXO#1送给Bob，UTXO#2留给自己，并把Commitment以OP\_Return操作码的形式写到比特币链上；

5.第4步完成后，再将第1步生成的CKB交易发送至CKB链上。



上面省略了一些比较复杂的细节。事实上，当Alice把自己的RGB++资产转移给Bob时，要先进行复杂的身份证明，证明自己的确是Cell#0 的主人。这里面涉及的事情，包括：

1.证明Cell#0 和BTCUTXO#0的确有绑定关系；

2.Alice证明自己是Cell#0和BTC UTXO#0的实际控制者。

要注意，写有RGB++资产数据的Cell和比特币UTXO可以被比特币账户同步改写，整个交互流程中，通过比特币账户即可完成一键式操作。上述场景不仅限于比特币和CKB之间的同构绑定，可以拓展到Cardano、Liquid、莱特币等广阔的范畴，想象空间还是很大的。

## Leap的实现原理与支持场景

前面我们曾提到，Leap功能实际就是切换RGB++资产绑定的UTXO，比如将其从比特币换绑到Cardano，之后就可以用Cardano账户控制RGB++资产。此后你还可以在Cardano链上转账，把控制RGB++资产的UTXO拆分转移给更多人。

通过这种方式，RGB++资产可以在多条UTXO公链上转移和分发，但却可以绕开传统跨链桥Lock-Mint的模式。在这个过程中，需要由CKB公链充当类似于索引器的角色，见证并处理Leap请求。

假设你要把BTC绑定的RGB++资产转移给Cardano账户，最核心的几步无外乎：

1. 在比特币链上发布Commitment，声明将BTC UTXO绑定的Cell解绑；

2. 在Cardano链上发布Commitment，声明将Cell绑定至Cardano UTXO；

3. 变更Cell的锁定脚本，将解锁条件关联的比特币UTXO，变为Cardano上的eUTXO。



我们可以注意到，在这整个流程中，RGB++资产数据仍然存放在CKB链上，只是把解锁条件关联的比特币UTXO，变更为Cardano链上的eUTXO。当然具体的执行流程比上面说的复杂不少，在此不赘述。

此外在leap方案中有一个隐性前提，即CKB公链作为一种第三方的见证人、索引以及DA设施。作为公链其可信度要远超传统跨链桥的MPC和多签等方式。

其实基于Leap功能还可以实现很有意思的场景，比如我们可以实现“全链交易”。假设我们横跨比特币、Cardano和CKB搭建起索引器，构建一个交易平台，允许买家和卖家交易RGB++资产，买家可以把自己的比特币转给卖家，然后用自己的Cardano账户接收RGB++资产。

这个过程中，RGB++资产的数据还是记录在Cell中，但这个Cell会被转移到买家手中，然后其解锁权限从卖家的比特币UTXO变更为买家的Cardano eUTXO。

## Wrapper

虽然Leap功能对于RGB++资产是完美的，但还是有一些瓶颈：

对于比特币和Cardano而言，RGB++资产本质是基于OP\_RETURN操作码的铭文/符文/染色币。这些公链节点无法感知到RGB++资产的存在，而CKB实际上是以索引器的身份从中参与协调。也就是说，对于比特币和Cardano而言，RGB++ Layer主要支持的是铭文/符文/染色币的Leap，而不是BTC、ADA等原生资产的跨链。

对此，RGB++ Layer官方引入了Wrapper，可以简单理解为一种基于欺诈证明和超额质押的桥。以rBTC wrapper为例，它将BTC桥接到RGB++ Layer，在RGB++ Layer上运行的一组智能合约会监控桥的守护者。如果守护者有恶意行为，他们的抵押物将被slash。如果守护者串通盗窃锁定的BTC，rBTC持有者可以获得全额赔偿。



在结合了Leap和Wrapper后，BTCFi生态中的各种资产如RGB++原生资产、BRC20、ARC20、符文等都可以跨到其他层或公链上去。



下图是应用LeapX的使用流程的一部分，可以看到它几乎支持了所有BTCFi主流资产到不同生态体系的互操作性。并且对不同发行方式的资产都有相应的相应的处理流程，有一部分用的wrapper，有一部分用的是leap。



## CKB-VM：BTCFi的智能合约引擎

上面我们主要解释了RGB++ Layer的同构绑定与Leap概念。下面让我们考察其他的要点。

在传统的BTCFi中，由于缺乏智能合约的支持，只能实现一些比较简单的Dapp，有些实现方法会有一定中心化风险，有些则比较笨拙不灵活。

为了实现在区块链上可用的智能合约层，CKB为RGB++ Layer提供了CKB-VM，任何能够支持RISC-V虚拟机的编程语言都可以用于在RGB++ Layer上进行合约开发。开发者可以使用他们偏好的工具和语言，在统一的智能合约框架和执行环境下，实现高效、安全的智能合约的开发与部署。

以下是一段用C语言实现的CKB中用户自定义代币UDT的transfer方法。可以看到除了语言不同，其基础逻辑和一般的代币都是相同的。而由于RISC-V有广泛的语言和编译器支持，对开发者的智能合约开发入门要求就比较低，我们可以很轻松的用JavaScript、Rust、Go、Java 和 Ruby把这段逻辑重写出来，而非必须学习某种DSL语言才可以编写合约。

当然，语言只是编程的一个方面，具体的智能合约框架的学习是不可避免的。



## 原生AA生态：无缝衔接BTC与RGB++

最后让我们再简单了解下RGB++ Layer背后的原生AA与账户抽象生态。由于BTCFi本质是为原生的比特币资产提供多样性的Defi体验，能否兼容主流比特币钱包将会是BTCFi周边设施需要考虑的重要因素，而RGB++ Layer直接复用了CKB的原生AA方案，可以在可开发者侧和用户侧都尽量与BTC和Cardano等重要的UTXO公链兼容。

在RGB++ Layer中，用户可以使用不同的签名算法进行鉴权。如，用户可以使用BTC、Cardano甚至WebAuthn等账户、钱包或鉴权方式，直接操纵RGB++ Layer上的资产。

我们以下面的钱包中间件CCC为例，它可以为钱包和dApp提供各种公链对CKB的可操作性。

下图是CCC的连接窗口。我们可以看到，它支持Unisat和Metamask等主流钱包入口。



另一个例子是WebAuthn的实现，CKB生态钱包JoyID就是典型代表。通过 JoyID，用户可以直接通过生物识别（如指纹或面部识别）方式进行身份验证，实现无缝且高安全性的登录和身份管理。

可以说，同构绑定和Leap能够成立的基础就在于RGB++ Layer拥有完备的原生AA方案，可以很好的兼容其他公链的账户标准，这种特性不但便于支持一些关键场景，同时也可以为UX扫清障碍。

## 总结

在上文中，我们对RGB++ Layer的全貌进行了考察，它可以作为铭文/符文/染色币等各种Memecoin的重要基础设施，实现全链交互的场景。而RGB++ Layer基于RiscV构建的智能合约执行环境，可以为BTCFi所需要的复杂业务逻辑创造土壤。

碍于篇幅所限，本文只是对RGB++ Layer核心技术的简单科普，并没有对许多复杂的细节进行系统性的科普。未来我们将持续关注RGB++ Layer的进展，对该项目相关的一系列技术方案进行更为透彻和深度的解析，大家敬请期待！