**在用户-浏览器阶段，采用ABY3秘密共享协议。**将密码分成三份秘密份额，分别存储在用户本机、浏览器和一个半诚实（诚实但好奇）的第三方中，只有获取其中的至少两份秘密份额，用户才可以复原出登录密码，并将密码传输给Web服务器。其中半诚实第三方可以是我们的密码管理系统第三方，也可以是属于用户的其它物理设备。在我们的设计之中阐述的第三方为我们的密码管理系统。

如果用户对我们的系统云端作为半诚实第三方存储感到怀疑或不信任，我们也**提供了基于Cheetah协议的半诚实两方模型**。Cheetah协议会将用户个人隐私密码仅仅划分为两个秘密份额，分别存储在用户本机与浏览器之中，完全保证用户掌控自己的密码隐私。

Shamir于1979年提出了基于拉格朗日(Lagrange)插值定理的(t,n)门限方案。该方案利用有限域上的n次随机多项式来分享秘密，被分享的秘密为多项式的零次系数，恢复秘密至少需要t个多项式上的点。为了提高系统的完整性并完善系统功能，我们进一步设计了数据泄露溯源模块。**该模块的主要目的是解决在发现分布式服务器中存储的秘密份额泄露时，如何确定是哪台服务器出现了问题导致数据泄露的情况。**

我们选择其中一台服务器作为主服务器。当我们注册密码后，首先会将密码的原文重定向发送到主服务器，主服务器会计算出 n 个秘密份额，并将这些秘密份额与额外的信息进行拼接，然后发送给其他 n-1 台服务器。在这种情况下，拼接节点编号或节点哈希值的算法仅部署在主服务器上，这符合 Semi2k-SPDZ 协议中一方作为可信方的要求。

在检测时我们不需要完成次排列组合的秘密恢复，我们只需要成功完成一次的秘密恢复即可，当系统完成了一次的秘密恢复后，从成功恢复的组合之中选取 t-1 个秘密份额，再将该 t-1个秘密份额分别与其余的 n-t个秘密份额逐一进行秘密恢复尝试，如果未能恢复成功，那么新加入的一方即为恶意参与方。

Semi2k：Semi2k-SPDZ协议能够在协议运行过程中检测到恶意方并停止运行，以防止数据泄露。然而，要在协议中进行恶意方检测，需要使用我们优化后的组合方法，该方法适用于恶意方出现频率较低的情况。对于恶意方出现频率较高的情况，我们建议集成基于可检测恶意方的NPC协议，以进一步增强安全性。通过引入这些安全机制，我们能够有效应对潜在的恶意方，并提供更可靠的数据保护。

Additive Secret Share方案实现了对秘密份额在其自身服务器上定时刷新，来达到秘密份额不是一成不变的效果。我们也能够选择定时收集所有秘密份额并重新划分，但是这会带来极大的性能开销。

Brickell方案是对Shamir方案的改进，它将一维方程扩展为多维向量方程，运用多项式插值方法。该方案在数据安全方面表现出较好的性能，能够在不泄露重建秘密的情况下，检查参与者的行为是否一致，并支持动态增减参与者，即使不知道最终参与者的个数。然而，由于其在加密和解密过程中的计算开销较大，相对于Shamir方案而言，效率会有所降低。通过引入Brickell秘密共享方案，我们能够灵活应对服务器数量的动态变化，在保证数据安全性的同时提供更强的灵活性