ASIACCS'22-Feng Hao - Sok: Password-Authenticated key Exchange - Theory, Practice, Standardization and Real-World Lessons

相关述：(S)自EKE提出以来，PAKE 30年在理论、实践、标准化等方面都取得了一定的进展，但引出了一系列的问题，例如为什么PAKE在实践中应用较少，这些问题鲜有人讨论，而且也没有对PAKE的历史做系统介绍的研究，(T)作者希望对上述问题做出回应，(A)通过系统研究过去30年的PAKE协议，(R)根据属性和设计策略对协议进行了分类，对不同类型的代表协议做了性能对比，分析了PAKE在实际应用以及与非PAKE方案对比的优缺点。

优点：① 对PAKE过去30年的发展进行了详细的归纳、总结与分析，比较全面细致
② 在比较性能时引入了之前研究通常会忽略的 group setup 和 exponent length
③ 对部分协议又难以在实践中使用的原因进行了分析。

问题：① 文章对安全性未做比较，可对代表的方案是否抵抗常见攻击手段做分析
② 本文未结合口令认证领域的其它内容讨论，如 Password Hardening 等，可简要讨论
③ 本文的论述如 PAKE分为5类，缺少一定的合理性论述。

▲PAKE研究引出的几个问题(仅为例证，不全面)
1> 一些PAKE协议只包含2~3个flow，但为什么难以正确处理?
2> 为什么提出的可证明安全的协议很少在实践中使用?
3> 为什么标准化的PAKE协议仍可能存在漏洞?
4> PAKE协议可以抵抗钓鱼攻击吗?为什么没有替换web应用中的口令认证?

▲ 3个时间阶段:
1> 1992-2008，IEEE P1363.2，使用8年的时间测试PAKE在实践中的应用，但几乎均存在安全问题，最终只有SRP-6a和 Patched SPEKE在实践中使用.
2> 2008-2018，ISO/IEC 11770-4标准修订了J-PAKE、AugPAKE和SPEKE的一个修订版本，另外，PAKE也逐步用于更多应用中，如 SRP-6a 用于1password的凭证恢复。
3> 2008-2021(至写作时间)，IETF进一步推进发展，特别是将PAKE集成到TLS中. 另外评定了标准的PAKE协议，包括CPace和OPAQUE.

▲ PAKE可以分为两大类
1> balanced PAKE：双方共享一个secret(password或其生成值)，通常包含的安全需求：①抵抗离线字典攻击 ② 限制攻击者1次执行只能猜测1次口令 ③ sk的安全性 ④提供前向安全性
若存储在server的secret丢失，则攻击者可伪装成client，引出了server只存储password的单向转换.

2> Augmented PAKE：除了上述4个要求，还包括⑤server compromise resistance 和⑥pre-computation resistance. 这两个措施只增加了攻击的负担，但无法阻止攻击

▲由于公钥通常具有特定的代数结构，因此使用口令d加密公钥可能存在问题，也因此几乎所有的实践PAKE设计者限于离散对数而非大因数分解.

DH-EKE的安全性由BRP三人证明，采用的 BRP model也被后续研究经常使用.(EKE2).
对EKE2的证明使用Ideal Cipher，但并未给出实现方式，故无法实践使用.

▲PAKE中口令的三种用途：①作为加密密钥 ②用于生成生成元 ③作为模运算中的整数
▲分类考虑了安全模型，假设和具体的构造方法

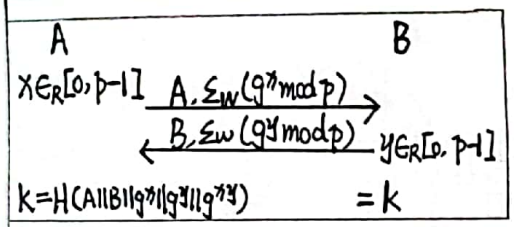
1. PAKE的安全证明通常使用以下模型中的1种或多种
1> Ideal Cipher(IC)：对称加密算法不会泄漏出任何明文的信息(即使是低熵的key)
2> Common reference string(CRS)：返回所有人信任的CRS的函数，例如多个生成元，但其离散对数关系未知
3> Random Oracle(RO)：返回定长的随机串的函数，许多在ROM中证明安全性的PAKE协议会使用hash-to-group(H2G)或hash-to-curve(H2C)，生成一个非单位元的随机元素.
主要考虑以下两种设定：multiplicative group over a finite field(MODP)
additive group on an elliptic curve(EC)

2. 分类
1> Password as an encryption key：即使用口令作为加密密钥，需使用ideal cipher假设，例如EKE和EKE2等，此类协议由于难以实现ideal cipher而无法广泛使用。
2> Password-derived generator：协议的群的生成元由口令生成，例如Patched SPEKE使用H2G函数将口令映射成群的生成元；PACE v2则是用H2C将口令映射到椭圆曲线的点上，本方案的挑战是实现H2G或H2C的函数f，IEEE 1363.2实现了此函数，但由于不是常数时间运算会引侧信道攻击；IETF在着手解决
3> Trusted Setup(TS)：定义两个或多个生成元，它们之间的离散对数关系未知，例如KOY，本身是希望不使用ROM证明PAKE的安全性，由于依赖于TS，证明依赖于CRS模型(但也可依赖于ROM和1CW)，SPAKE2为此类最高效的。但此类方法的问题为，无法合理实现TS，当前没有主要的商业部署基于TS
4> Secure two-party computation：PAKE被视为在equality function上的双方安全计算问题，使用non interactive ZKP检查实体是否规范执行，例如J-PAKE.
5> Password-derived exponent，server存储g^w，例如SRP-3、SRP-6a

▲性能对比中，由于模幂运算与指数的bit长度呈线性关系，因此在不同的器件中，实际耗时运算不仅是流程图的次数
▲提到PAKE在现实中使用的两个基本需求：
1> 可信的口令输入接口(在PAKE中通常集成到底层OS或应用中，而webpage不够安全，故PAKE很少用于web应用
2> 用于双方共享口令的可信带外通道
在Credential Recovery, Device pairing和E2E secure channel三种应用场景与Non-PAKE方案对比

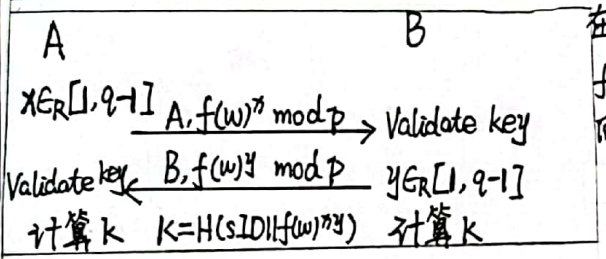
▲ Lessons：①设计的PAKE给出完整且公开的说明 ②合理的假设和底层模型 ③不断地修订标准 ④新的使用环境 ⑤对安全模型、性能及其它功能进行系统化的权衡，用于比较.

EKE2 Protocol (Class 1)



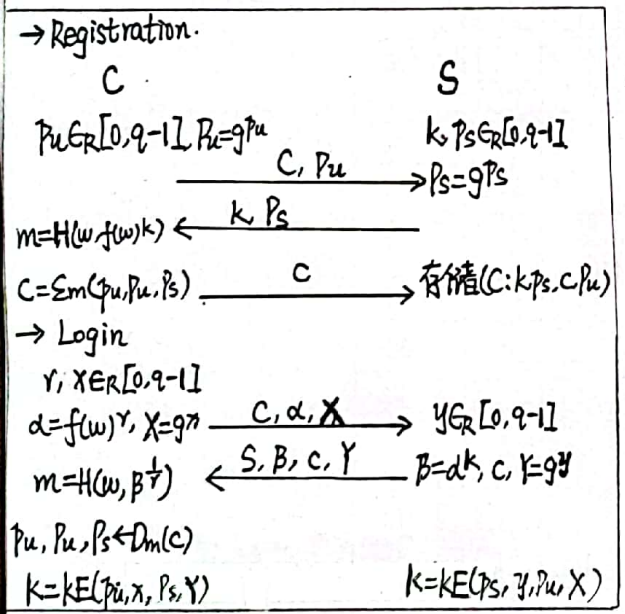
A和B各执行2次模幂运算
1> 临时公钥
2> 会话密钥
无法直接转换为其它的群，如EC、DSA

Patched SPEKE Protocol (Class 2)



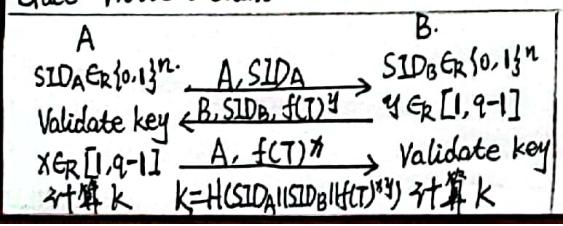
在Z_p^*的子群上运行，所为q有$p = 2q+1$
$f(w) = H(w)^2 \bmod p$
同样A和B执行2次模幂运算.

OPAQUE Protocol (Class 2)



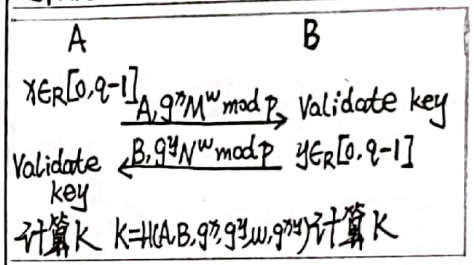
在Z_p^*的所为q的子群中运行，$q|p-1$.
模运算 mod p
KE为AKE方案，推荐HMQV,
$d = H(X, s)$, $e = H(Y, u)$, 可输出为$\lfloor \log_2 q \rfloor / 2$ bit
$k = H(g^{(x+da_u)(y+eP_s)})$

Cpace Protocol (Class 2)



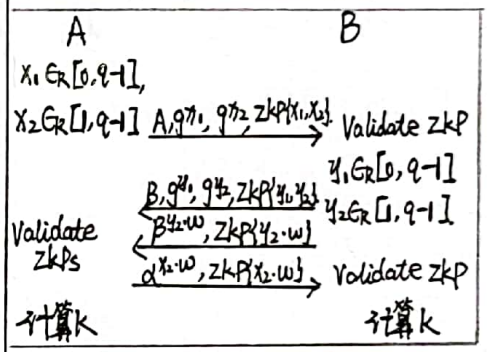
在Z_p^*的所为q的子群运行，$q|p-1$
$T = H(SID_A\|SID_B\|w\|A\|B)$.
与Patched SPEKE计算形式相同

SPAKE2 Protocol (Class 3)



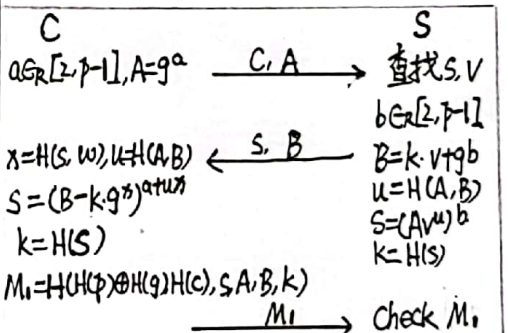
trusted setup 包括 {g, M, N} 群运算同上
A有3次模幂运算：
① 计算 $g^x M^w$.
② 验证 $g^y N^w$
③ 计算 g^{xy}

JPAKE Protocol (Class 4)



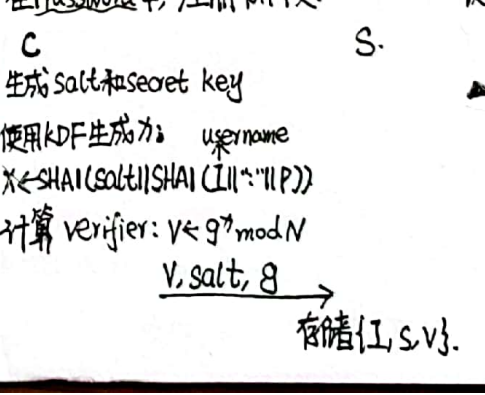
群运算同上，
$\alpha = g^{y_1} g^{x_2} g^{x_1}$, $\beta = g^{x_1} \cdot g^{x_2} \cdot g^{y_1}$
计算 $U = \alpha^{x_2 \cdot w}$, $V = \beta^{x_2 \cdot w}$.
Alice计算：$K = H(C(V / g^{x_2 \cdot y_2 \cdot w})^{x_2}) = H(g^{(x_1+y_1) \cdot x_2 \cdot y_2 \cdot w})$
Bob 计算 $k = H((U / g^{x_2 \cdot y_2 \cdot w})^{y_2}) = H(g^{(x_1+y_1) x_2 \cdot y_2 \cdot w})$
Alice 有4次模幂运算：
3次运算 g^{x_1}, g^{x_2}, $\alpha^{x_2 \cdot w}$.
3次计算3个ZKP, 6个验证ZKP, 2个计算 sk.

SRP-6a Protocol (Class 5)



在Z_p^*上运算，注册阶段 S存储S和 $V = g^{H(S,w)}$.
Client 3次模幂 g^a, g^x, S
Server 3次模幂 g^b, V^u, S.

在1Password中，注册阶段：



认证阶段，x仍采用KDF方式，其中有Secret key
Secretkey 由Two-Secret key Derivation 生成
在1Password中，Secret key 为34个字母-数字组成的存储在登录的设备中，
它使敌手获取verifier时，若未知Encrypt key难以猜测口令。另外，KDF采用