USENIX'17 - Lai - Phoenix: Rebirth of a Cryptographic Password-Hardening Service

优点：① PH提出的背景介绍较好，本文研究动机清晰

② 指出 Schneider 等人方案的错误，并详细进行了说明

③ 方案的原理，设计，安全性分析和性能评估都进行了说明

问题：① crypto server 可知登录用户的用户名 un，泄漏了用户的登录行为

② 本文中仅有一个 crypto server，存在单点故障的问题，可用性受损（未来工作分析式）

③ 对于 web service、crypto server compromise，不能及时检测该类事件

④ key rotation 部分应区分 service 和 server compromise，否则 if service 要更新，记录也更新，不合理 server key 也更新

External Password Hardening Services 使用 crypto server 执行特定的密码操作（如 PRF），在无需客户端更改的同时使 web service 无法独立验证口令的正确性。本文指出先前研究存在的问题（PYTHIA 仅在强假设安全且 pairing 耗时较大，Schneider 的方案有离线字典攻击的危险），提出有更正确标准的安全定义（如需要 key rotation）的 PHOENIX 安全性更高，且更高效。

Crypto Password Hardening (PH)　C: web service　S: crypto server.

→ 定义：PH 是一个双方协议，包含算法 (Setup, kGen$_C$, kGen$_S$, ⟨C,S⟩enrl, ⟨C,S⟩vrf, ⟨C,S⟩rot, Udt)，包含4个阶段：

① Setup Phase：C与S在无交互的情况下独立设置 public key 与 secret key.

▲ 输入安全参数 λ，$pp \leftarrow \text{Setup}(1^\lambda)$，C: $(pk_C, sk_C) \leftarrow \text{kGen}_C(pp)$，S: $(pk_S, sk_S) \leftarrow \text{kGen}_S(pp)$

② Enrollment Phase：运行协议 ⟨C(sk$_C$, un, pw, aux), S(sk$_S$, un, aux)⟩enrl，

▲ S无输出，C输出 enrollment record T，T包括(un, pw)，C存储 (T, un).

运行后 C 删除 pw，并且 C与S删除所有运算中间值.

③ Validation Phase：运行协议 ⟨C(sk$_C$, T, un, pw), S(sk$_S$, un)⟩vrf.

▲ S无输出，C输出 $b \in \{0,1\}$ 以标识 T 是否存储着 (un, pw).

④ key Rotation Phase：运行协议 ⟨C(sk$_C$), S(sk$_S$)⟩rot，

▲ S输出更新的密钥对 pk_S', sk_S'，C输出更新的密钥对 pk_C', sk_C' 及 update token τ 来更新记录

C运行更新算法 Udt(τ, T, un)，更新所有的注册记录.

PH安全性分析（思路）

▲ Partially Oblivious：恶意的S无法获知 pw，但 un 是S已知的.

C: challenger，S: adversary　（更新后的）

C生成 sk$_C$ 并仿真所有的协议，$\mathcal{A}$ 除了 sk$_C$ 之外其余的输出均可获得.

① learning phase：$\mathcal{A}_1$ 与C运行协议，最终输出 un*, pw$_0$*, pw$_1$*

② challenge phase：C与 $\mathcal{A}_2$ 共同生成 challenge record T*，最终 $\mathcal{A}_2$ 输出 T* 对应 pw$_b$* 的 b'

（其中若 $\mathcal{A}$ query 了 (un*, pw$_{1-b}$*) 返回 $\perp$），$|\Pr[\text{Obl}^0_{\pi,\mathcal{A}}(1^\lambda)=1] - \Pr[\text{Obl}^1_{\pi,\mathcal{A}}(1^\lambda)=1]| \le \text{negl}(\lambda)$

▲ Hiding：否破坏C的数据可获取 sk$_C$，注册记录. 若想获取 pw，$\mathcal{A}$ 至多可以通过在线猜测得 pw.

C: adversary，S: challenger.

S为 $\mathcal{A}$ 提供 magical oracle，给定 guess，返回是否与 pw 相等，若允许 Q 次猜测，则至多猜测前 Q 个最可能的口令.

① S生成 sk$_S$，$\mathcal{A}_1$ 与S交互，最终输出 sk$_C$，un* 及口令的分布 $\mathcal{X}$

② S选取 $pw^* \xleftarrow{\$} \mathcal{X}$，并计算一个 T^* (un*, pw*)，该值由正确诚实的 C 产生），发送给 $\mathcal{A}$，

③ $\mathcal{A}_2$ 与S交互并最终输出 pw'，若 pw' = pw* 则 $\mathcal{A}$ 获胜.

假设 p_i 是第 i 最可能的口令，$\Pr[\text{Hiding}_{\pi,\mathcal{A}}(1^\lambda)=1] \le \sum_{i=1}^{Q+1} p_i + \text{negl}(\lambda)$.

▲ Binding：恶意的S无法欺骗C T对应 (un, pw) 及 (un', pw')，保证 (un, pw) 只对应唯一合法的 enrollment record，注意 record 只用于验证而不会公开.

C: challenger，S: adversary

① $\mathcal{A}_1$ 输出 sk$_C$，T^* 及 (un*, pw*)，C与 $\mathcal{A}_2$ 交互并验证 (T^*, un*, pw*).

② $\mathcal{A}_3$ 生成另一对 (un*, pw*)，C与 $\mathcal{A}_4$ 交互验证 (T^*, un*, pw*).

③ 若①、②均验证通过，$\mathcal{A}$ 获胜.

$\Pr[\text{Binding}_{\pi,\mathcal{A}}(1^\lambda)=1] \le \text{negl}(\lambda)$

▲ Forward Security：$\mathcal{A}$ 无法区分更新的 key, record 与新生成的值

C和S: challenger，$\mathcal{A}$ 为外部敌手.

① $\mathcal{A}_1$ 输出 sk$_C$ 和 sk$_S$ 及合法的 (T, un, pw).

② C与S更新 sk$_S$, sk$_C$，enrollment record 或生成新的 key 和 record，发送给 $\mathcal{A}$.

③ $\mathcal{A}_2$ 输出得到的值是如何产生的

L 为将 T 映射到辅助信息 aux 的 leakage function，$|\Pr[\text{Rot}^0_{\pi,\mathcal{A},L}(1^\lambda)=1] - \Pr[\text{Rot}^1_{\pi,\mathcal{A},L}(1^\lambda)=1]| \le \text{negl}(\lambda)$

PHOENIX 基于 (partially) homomorphic encryption 及 pseudorandom function.

▲ 有限循环群 $\mathbb{G}$，阶 $q = q(\lambda)$. $H_{rc}: \{0,1\}^\lambda \times \{0,1\}^{n_s} \to \mathbb{G}$

π 为 non-interactive zero-knowledge proof of knowledge system.

▲ Setup Phase.

Setup(1^λ)：$crs \xleftarrow{\$} \pi.\text{Gen}(1^\lambda)$，$g \xleftarrow{\$} \mathbb{G}$.

kGen$_C$(pp)：$pk_C \leftarrow \perp$，$sk_C \leftarrow k_C \xleftarrow{\$} \mathbb{Z}_q$

kGen$_S$(pp)：$s, x, y, k_S \xleftarrow{\$} \mathbb{Z}_q$，$h \leftarrow g^s$，$z \leftarrow g^x h^y$，$pk_S \leftarrow (h, z)$，$sk_S \leftarrow (s, x, y, k_S)$

▲ Enrollment Phase

输入的辅助信息 aux 要么为空串 ϵ，要么为 (n_S, n_C) 用于证明简洁安全性.

C S

aux≠ε时，aux=(ns, nc) ...

否则：$n_c \leftarrow \{0,1\}^\lambda$ ... $n_s \leftarrow \{0,1\}^\lambda$

$r \leftarrow \$ Z_q$, $h_c \leftarrow Hc(un, pw, nc)^{kc}$ $h_s \leftarrow Hs(un, n_s)^{ks}$.

$\xleftarrow{h_s, n_s}$

返回 $T \leftarrow (g^r, h^r \cdot h_s \cdot h_c, Z^r, n_s, n_c)$ Z^r用于证明hiding属性.

▲Validation Phase. C验证T是否与给定的un, pw对应.

C S.

$T := (T_1, T_2, T_3, n_s, n_c)$.

$u \leftarrow \$ Z_q$, $b \leftarrow 0$ 同态

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \leftarrow \begin{bmatrix} T_1 \cdot g^u \\ T_2 \cdot h^u / Hc(un, pw, nc)^{kc} \\ T_3 \cdot Z^u \end{bmatrix} \xrightarrow{(C_1, C_2, C_3, n_s)} \begin{bmatrix} C_2 \\ C_3 \end{bmatrix} \stackrel{?}{=} \begin{bmatrix} C_1^s \cdot Hs(un, n_s)^{ks} \\ C_1^r (C_2 / Hs(un, n_s)^{ks})^y \end{bmatrix}$$ 有

$\pi \leftarrow \$ \Pi \cdot Pok\{(s, k_s): C_1 = C_1^s \cdot Hs(un, n_s)^{k_s} \wedge h = g^s\}$

$\xleftarrow{\pi}$

$b \leftarrow \Pi \cdot Vf((g, h, C_1, C_2, Hs(un, n_s)), \pi)$.

返回b.

▲key Rotation Phase. 更新后仍可验证.

C S

$\alpha, \beta, \gamma, \delta \leftarrow \$ Z_q$.

$S := \delta + \alpha \cdot \eta \cdot S + \beta(y + \eta)$.

$\xleftarrow{\alpha, \beta, \gamma, S}$

$kc' \leftarrow \alpha \cdot kc$ $k_s' \leftarrow \alpha \cdot k_s + \gamma$, $s' \leftarrow \alpha \cdot s + \beta$

$pkc' \leftarrow \perp$ $x' \leftarrow \alpha \cdot x + \delta$, $y' \leftarrow y + \eta$

$skc' \leftarrow kc'$ $pk_s' \leftarrow (h^\alpha \cdot g^\beta, Z^\alpha \cdot g^\delta)$

$T \leftarrow (\alpha, \beta, \gamma, S)$ $sk_s' \leftarrow (s', x', y', k_s')$

返回(pkc', skc', T) 返回(pk_s', sk_s').

Udt(T, T, un)

$g_6 \leftarrow Hs(un, n_s)$

$V \leftarrow \$ Z_q$

$T_1' \leftarrow T_1 \cdot g^V$

$T_2' \leftarrow (T_2 \cdot h^V)^\alpha \cdot (T_1 \cdot g^V)^\beta \cdot g_6^\gamma$

$T_3' \leftarrow (T_3 \cdot Z^V)^\alpha \cdot (T_1 \cdot g^V)^S$

返回. $T' \leftarrow (T_1', T_2', T_3', n_s, n_c)$

Π的实例化

▲$\Pi.Gen(1^\lambda)$: $H \leftarrow \$ \mathcal{H} = \{H: \{0,1\}^* \to Z_q\}$, 返回 $crs := H$

▲$\Pi.Vf((g, h, C_1, C_2, g_s), \Pi)$, 其中$\Pi$为$(\bar{h}, \bar{C}_1, \bar{g}_s, \bar{s}, \bar{k}_s)$:

$c := H(g, h, C_1, C_2, g_s, \bar{h}, \bar{C}_1, \bar{g}_s)$

$b_1 := (C_1^{\bar{s}} g_s^{\bar{k}_s} \stackrel{?}{=} \bar{C}_1 \cdot \bar{g}_s \cdot C_2^c)$, $b_2 := (g^{\bar{s}} \stackrel{?}{=} \bar{h} \cdot h^c)$

返回 $b := (b_1 \wedge b_2)$

▲$\Pi.Pok\{(s, k_s): C_2 = C_1^s \cdot g_s^{k_s} \wedge h = g^s\}$

$r_1, r_2 \leftarrow \$ Z_q$, $\bar{h} := g^{r_1}$, $\bar{C}_1 := C_1^{r_1}$

$\bar{g}_s := g_s^{r_2}$, $c = H(g, h, C_1, C_2, g_s, \bar{h}, \bar{C}_1, \bar{g}_s)$

$\bar{s} := r_1 + c \cdot s$ $\bar{k}_s := r_2 + c \cdot k_s$.

返回 $\Pi := (\bar{h}, \bar{C}_1, \bar{g}_s, \bar{s}, \bar{k}_s)$

安全性分析(CDDH, DL, ROM) (2)

▲Partial Oblivious: 将 $T_2^* = Hc(un^*, pw_\delta^*, nc^*)^{kc}$ ($b \in \{0,1\}$)用真随机值替换，后可支持pw_0, pw_1

$Exp_{b,0}$：与$Obl^b_{\Pi,\mathcal{A}}$相等

$Exp_{b,1}$：challenger仿真Hc(un, pw, nc)，取$a \leftarrow \$ Z_q$，返回$g^a$，与$Exp_{b,0}$功能相同

$Exp_{b,2}$：由于aux=ε，若之前询问过nc^*，则challenger abort，发生概率$O(2^{-\lambda})$，否则与$Exp_{b,1}$不可区分

$Exp_{b,3}$：给定challenger一个extended DH-tuple$(g, g^{kc}, g^r, g^\theta, g^\delta, g^\varepsilon)$，其中$\delta = kcr$，$\varepsilon = kc\theta$

challenger未知kc，无法直接计算$Hc(un, pw, nc)^{kc}$，通过$(g^a)^{kc} = (g^{kc})^a = Hc(un, pw, nc)^{kc}$

当收到$\mathcal{A}$的$(un^*, pw_\delta^*, pw_{1-\delta}^*)$时，challenger计算：

$Hc(un^*, pw_\delta^*, nc_\delta^*) := g^r$, $Hc(un^*, pw_\delta^*, nc_\delta^*)^{kc} := g^\delta$, $Hc(un^*, pw_{1-\delta}^*, nc_{1-\delta}^*) := g^\theta$, $Hc(un^*, pw_{1-\delta}^*, nc_{1-\delta}^*)^{kc} := g^\varepsilon$

与$Exp_{b,2}$功能相同

$Exp_{b,4}$：给定challenger一个随机tuple$(g, g^{kc}, g^r, g^\theta, g^\delta, g^\varepsilon)$，$\delta, \varepsilon \leftarrow \$ Z_q$.

仍按$Exp_{b,3}$中仿真Hc，则由DDH假设，与$Exp_{b,3}$不可区分.

可知$Exp_{0,4}$和$Exp_{1,4}$功能上相同，则有$Obl^0_{\Pi,\mathcal{A}}$与$Obl^1_{\Pi,\mathcal{A}}$计算上不可区分.

▲Hiding: $q_T \geq 2^\lambda$，将challenge enrollment record转换为随机值的client为$\mathcal{A}$.

Exp_0：与$Hiding_{\Pi,\mathcal{A}}$相同

Exp_1：使用Π的simulator仿真，与Exp_0计算上不可区分

Exp_2：仿真Hs，$\mathcal{A}$询问Hs(un, ns)时，$\gamma \leftarrow \$ Z_q$，计算$Hs(un, ns) := g^\gamma$.

后续执行enroll时，nc_δ^*, ns^*均随机选取并计算$Hs(un_\delta^*, ns^*)$.

若之前询问过相同的ns^*，则abort，概率为$O(2^{-\lambda})$，否则与Exp_1计算上不可区分

Exp_3：给定challenger一个DH-tuple$(g, g^\delta, g^{ks}, g^\eta)$，$\gamma = \delta \cdot k_s$.

设置pp并计算pk_s，$\mathcal{A}$请求un^*的记录

challenger计算$Hs(un^*, ns^*) := g^\delta$并用$g^\eta$代替$Hs(un^*, ns^*)^{ks}$.

只有$\mathcal{A}$之前请求过ns^*作为输入的值能区分Exp_3可能，但Exp_2已排除..

Exp_4：给定challenger一个随机tuple$(g, g^\delta, g^{ks}, g^\eta)$，其中$\eta \leftarrow \$ Z_q$，按照$Exp_3$仿真Hs.

由DDH假设，Exp_4和Exp_3在计算上不可区分.

Exp_5：考虑验证时段仅满足C_3的等式，不满足C_2. (C2中无s) 由$Z = g^x h^y \Rightarrow \log_g Z = x + sy$

x和y不是唯一确定的，猜对概率为$\frac{1}{q}$.

若高于$\frac{1}{q}$，则只满足C_3而非C_2，即$C_2 \neq C_1^s \cdot h_s$，令$s' \neq s$，$C_2 = C_1^{s'} \cdot h_s$，$C_3 = C_1^x (C_2/h_s)^y = C_1^{x+s'y}$

故有：$\log_{C_1} C_3 = x + s'y$.. 又由$s' \neq s$，故与$\log_g Z = x + sy$线性独立，故可区分x和y的值，

与事实违背，故发生概率不超过$\frac{1}{q}$.

Exp6：给定challenger一个DH-tuple (g, g^r, g^s, g^u)，$u=rs$.

替换challenge record为 $(T_1^*, T_2^*, T_3^*) = (C_1^* \cdot Hc(un,pw,nc)^{kc}, C_2^*, C_3^*)$.

其中 $(C_1^*, C_2^*, C_3^*) := (g^r, g^{u+\eta}, g^{rx+uy})$ ，与Exp5相同

Exp7：给定challenger一个随机tuple (g, g^r, g^s, g^u)，$u \leftarrow \$ Z_q$. 按Exp6仿真记录.

由DDH假设，与Exp6不可区分

Exp8：challenger 取 $r, s \leftarrow \$ Z_q$，$u \leftarrow \$ Z_q \setminus \{rs\}$，除去可忽略的概率，与Exp7不可区分

Exp9：$\mathcal{A}$ query 验证的oracle，并给S发送 (C_1, C_2, C_3, ns^*)

① 若 $(C_1, C_2, C_3) = (C_1^* \cdot g^v, C_2^* \cdot h^v, C_3^* \cdot z^v)$，存在v满足这等式，则敌手获胜

② 否则：challenger 输出 $\perp$.

与Exp8不可区分：

$\mathcal{A}$ 已知：$z = g^{rs+sy}$，$C_2^* = g^{u+\eta}$，$C_3^* = g^{rx+uy}$ 及对数形式 $\Rightarrow \log C_3^* - r\log z = (\log C_2^* - rs)y - \eta y$

上式为 (y, η) 的二次方程，故可能有2个解.

假设 $\mathcal{A}$ query 验证oracle，响应的 (C_1, C_2, C_3) 满足 $C_3 = C_1^y C_2^y g^{-\eta y}$

代入得 $\Rightarrow \log C_3 - \log C_1 \log z = (\log C_2 - \log C_1 \cdot s) \cdot y - \eta y$.

为了满足，$\mathcal{A}$ 需猜测出 (y, η) 或保持系数不变，故与Exp8不可区分.

Exp10：challenger 将 T_1^*, T_2^*, T_3^* 替换为随机值，与pw彼此独立.

除非 $\mathcal{A}$ 猜测 pw^* 正确，否则不可区别Exp9和Exp10，概率至多为 $\sum_{i=1}^{Q_{H_s}+1} p_i$

综上，$\mathrm{Hiding}_{\pi,\mathcal{A}}$ 的概率至多为 $\sum_{i=1}^{Q_{H_s}+1} p_i + \epsilon(\lambda)$.

▲ Binding 通过归约证明，S为 $\mathcal{A}$

PPT $\mathcal{A}$ 可破坏 Binding，则 PPT $\mathcal{B}$ 可解决DL问题.

假设 $\mathcal{A}$ 和 $\mathcal{B}$ 分别维护列表，$H_s(un, ns) := g_1^a$，$H_c(un, pw, nc) := g_2^b$

分析 $\mathcal{A}$ 获胜的情况①输出 $(sk_c, T^*, un_0^*, pw_0^*, state)$ 验证通过

$sk_c = kc$，$T^* = (T_1^*, T_2^*, T_3^*, ns^*, nc^*)$，有 a_0, b_0 使 $H_s(un_0^*, ns^*) = g_1^{a_0}$，$H_c(un_0^*, pw_0^*, nc^*) = g_2^{b_0}$

$\mathcal{A}$ 同样产生 (s, ks_0) 的proof π_0，有：$T_2^* = (T_1^*)^s \cdot H_s(un_0^*, ns^*)^{ks_0} \cdot H_c(un_0^*, pw_0^*, nc^*)^{kc}$，且 $h = g_1^s$

$= (T_1^*)^s \cdot g_1^{a_0 \cdot ks_0} \cdot g_2^{b_0 \cdot kc}$.

· $\mathcal{B}$ 可输出 (s, ks_0).

② 对 (un_1^*, pw_1^*)，同理，

故有 $g_1^{a_0 \cdot ks_0} \cdot g_2^{b_0 \cdot kc} = g_1^{a_1 \cdot ks_1} \cdot g_2^{b_1 \cdot kc} \Rightarrow \log_{g_1} g_2 = \dfrac{a_1 \cdot ks_1 - a_0 \cdot ks_0}{kc \cdot (b_0 - b_1)}$，$\mathcal{B}$ 解决DL问题

▲ Forward Security：证明 生成和更新 的key record 不可区分

→ $\mathrm{Rot}_{\Pi,\mathcal{A},\mathcal{C}}^b$，$\mathcal{A}$ 选取 secret key component，$(s, x, y, ks, kc) \in Z_q^5$.

对于 $(s', x', y', ks', kc') \in Z_q^5$ 和 $(\alpha, \beta, \gamma, \delta, \eta) \in Z_q^5$ 有一对一的对应关系

$$\begin{cases} s' = \alpha \cdot s + \beta \\ x' = \alpha \cdot x + \delta \\ y' = y + \eta \\ ks' = \alpha \cdot ks + \gamma \\ kc' = \alpha \cdot kc \end{cases} \qquad \begin{cases} \alpha = kc'/kc \\ \beta = s' - \alpha \cdot s \\ \gamma = ks' - \alpha \cdot ks \\ \delta = x' - \alpha x \\ \eta = y' - y \end{cases}$$

直接取和计算得来的 (s, x, y, ks, kc) 分布相同

→ $T = (T_1, T_2, T_3)$ 由 $\mathcal{A}$ 给出，格式为 $(g^r, g^{sr} \cdot g_s^{ks} \cdot g_c^{kc}, g^{(x+sy)\cdot r})$.

新记录 $T' = (g^{r'}, g^{sr'} g_s^{ks'} g_c^{kc'}, g^{(x+s'y')\cdot r'})$.

若 $b=0$，$r' = r + r$ $(r \leftarrow \$ Z_q)$. 若 $b=1$，$r' \leftarrow \$ Z_q$，两种情况给出的分布相同

评估

▲ Latency：考虑C与S之间的全部交互延迟.

▲ Throughput：PHOENIX比Schneider的方案每秒多处理50%的请求

比 PYTHIA ……，3倍 ……