TDSC'18 Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound(口令使用Zipf分布).

在当前2FA方案无综合系统的评估方法，而导致许多方案无法达到真正的2FA安全，也又注以做到安全性和可用性平衡的背景下，主要有以下3点贡献

1) 评估了6个2FA方案，提出了评估2FA的系统框架，包含苛刻的敌手模型和精炼的12个评估标准.

2) 引入了honeywords，并结合提出的fuzzy-verifier达到了方案安全性和可用性平衡.

3) 提出满足12个评估标准的方案并证明了安全性，并举例证明上述方案可以与其它2FA(甚至3FA)结合使用(包括单server和多server)

优点：① 指出当前2FA方案存在安全问题的原因，并绘制了发展流程图(Fig.2).

② 对敌手能力及评价标准的描述清晰，均解释了包含的原因.

③ 首次将honeywords 引入2FA协议设计的

问题：① 本方案采用nonce保证消息fresh，没有提到这一点，但评估中标记了C8，有些不妥

② 文中应简要说明honeywords可能带来的问题.

③ 存在部分笔误 1) 1.1第3段应为two factor scheme 2) 4.3 step V3为 $1-\frac{1}{n_0}$

真正的双因素安全(TDSC'15-Wang).

1) 拥有用户智能卡的敌手不能通过离线字典攻击恢复口令或伪装成用户.

2) 只获取用户口令但未得到smart card的敌手不能伪装成用户.

另外，方案也应该抵抗被动、主动攻击，并提供其它重要属性如用户匿名.

研究动机：

1) "fair comparison and general concensus are unlikely." + "there is no proper security justification". 如何通过其它领域的帮助实现真正的2FA安全.及 本地安全的口令更新.

2) 先前方案都假设口令均匀分布，严重低估了敌手的优势 本文使用更实际的Zipf分布

敌手模型：除了传统PAKE中的能力，smart card中的安全参数也可能被提取.

C-01：A可以离线字典. Did × Dpw (id和password空间的笛卡尔乘积).

C-02：A可以确定用户Ui的实体IDi

C-1：A可以控制整个通信通道.

C-2：A可以通过恶意card reader得到用户的口令，也可以通过侧信道攻击得到card中的数据，但无法同时做到(合理性分析). (1)

C-3：A可以获取先前的sk

C-4：当S被石破坏时，A可得到S的长期私钥及其它数据

▲ conditional non-tamper resistance assumption. 认为除非card被A长期持有，否则认为card仍是tamper-proof

评估标准(评估6个2FA方案+攻击实验，针对2FA应满足的安全性和用户友好性)

C1：S不存储口令验证表.

C2：口令易记忆且可由用户本地更改.

C3：口令不会被S管理员暴露.

C4：No smart card loss attack

C5：可以抵抗已知攻击.

C6：用户可以不更改实体信息撤销card.

C7：U和S可以构建sk

C8：无需时钟同步

C9：当用户误输入口令时，可及时及时提醒是

C10：U和S可实现双向认证

C11：可实现用户匿名性，保护用户身份，防追踪.

C12：提供前向安全性.

▲ 6个方案中，对于每条标准，至少5个满足，至少1个不满足，且每个方案至少有一条标准不满足

本文方案：4个阶段，有限循环群$G=\langle g\rangle$，素数阶q，长度l bit，G_1为Z*p的素数阶子群，$q|(p-1)$，Hash函数 $h_0:\{0,1\}^* \to \{0,1\}^{l_i}$ $i=0\sim3$. n_0代表(ID, PW)穷量大小 (x, $y=g^x \bmod p$)代表S的私钥和公钥，⇒代表安全信道，→代表普通信道

→ 注册阶段. $\in R_a$代表取随机值 (CT)

U → S

选IDi, PWi, $b \in R_a$ R1 $\xrightarrow{\{ID_i, A_i=h_0(b\|PW_i)\}}$ R2

时间T，选$a_i \in R_a$，→ $A_i=h_0(h_0(ID_i)\oplus h_0(b\|PW_i)) \bmod n_0$

Ui是否已注册？未注册，创建新条目，存储 {IDi, Treg=T, ai, Money-List=NULL}. R3.

已注册，则更新Treg，ai，并将Money-List置为0.

$N_i=h_0(b\|PW_i)\oplus h_0(x\|IDi\|Treg)$.

S → U

smart R4 card(SC). R5

{Ni, Ai, $A_i\oplus a_i$, q, g, y, n_0, $h_0(\cdot), \cdots, h_3(\cdot)$}.

Ui激活SC，SC要Ui输入两次b确保正确性.

其中，b可以为random，因为用户输入至SC后就可以忘记。可为128b此这与Treg均可抵抗暴力猜测，可满足C3.

→ 登录阶段 + 验证阶段.

U_i S

L1 输入SC，并输入 ID_i^*, PW_i^*，SC进步计算：

L2 $A_i^* \stackrel{?}{=} H_0(\cdots)$，验证 ID_i^*, PW_i^* 的正确性？

不等同，则 terminate

L3 SC选 $u \in R_a$，计算：

$C_1 = g^u \bmod p$, $Y_1 = y^u \bmod p$, $k = H_0(ID_i \| ID_s \| T_{reg}) = N_i \oplus H_0(b \| PW_i^*)$

$a_i = (A_i \oplus a_i) \oplus A_i$, $CID_i = ID_i^* \oplus H_0(C_1 \| Y_1)$

$CAk_i = (a_i \| K) \oplus H_0(Y_1 \| C_1)$ ← 注意

$M_i = H_0(Y_1 \| K \| CID_i \| CAk_i)$. L4: 发 $\{C_1, CID_i, CAk_i, M_i\}$.

抵抗 C-02.

→ 口令更改阶段（为满足C2，在本地进行）.

P1 U_i 输入SC，输入 ID_i 和 PW_i

P2. SC计算：$A_i^* \stackrel{?}{=} H_0(\cdots) \stackrel{?}{=} A_i$，否则 reject.

相等，证明 ID_i 和 PW_i 有效概率为：$1-\varepsilon$.

P3: SC要求 U_i 再提交 PW_i^{new}，计算：

$N_i^{new} = N_i \oplus H_0(b \| PW_i) \oplus H_0(b \| PW_i^{new})$.

$A_i^{new} = H_0((H_0(ID_i) \oplus H_0(b \| PW_i^{new})) \bmod n_0)$.

SC更新，N_i^{new}, A_i^{new}, $a_i \oplus A_i^{new}$.

V1 $Y_1 = (C_1)^y$, $ID_i = CID_i \oplus H_0(C_1 \| Y_1)$

验证 ID_i 格式是否正确，否则 terminate

V2 $k = H_0(b \| ID_i \| T_{reg})$

$M_i^* = H_0(Y_1 \| K \| CID_i \| CAk_i) \stackrel{?}{=} M_i$

不等则 terminate.

V3: $a_i' \| K' = CAk_i \oplus H_0(Y_1 \| C_1)$,

$a_i' \stackrel{?}{=} a_i$，否则 reject，成立则 A_i 正确

验证 $K' \stackrel{?}{=} k$，不等则计证明 $a_i' = a_i$，$K' \neq k$，表明 $1-\varepsilon$ 的概率 SC 被破坏

此时，S 执行：若 Honey-List 组织好 m_0 个，则将 K' 加入 List。若为 m_0 个，则 suspend 用户的 card，相等则继续.

V4: $\forall \in R_a$, $k_S = (C_1)^v \bmod p$, $C_2 = g^v \bmod p$

$C_3 = H_1(ID_i \| ID_S \| Y_1 \| C_2 \| K \| k_S)$

V5: 发 $\{C_2, C_3\}$.

V6: $k_U = (C_2)^u \bmod p$.

$C_3^* = H_1(CID_i \| ID_S \| Y_1 \| C_2 \| K \| k_U) \stackrel{?}{=} C_3 \Rightarrow S$ 合法.

$C_4 = H_2(CID_i \| ID_S \| Y_1 \| C_2 \| K \| k_U)$. V7: C_4 (C7) →

V8. $C_4^* = H_2(\cdots) \stackrel{?}{=} C_4 \Rightarrow$ Client 合法，且 accept，否则 terminate.

<--- V9 $SK = H_3(ID_i \| ID_S \| Y_1 \| C_2 \| K \| k_U)$. --->

十、协议设计依据.（实现真正的2FA安全）.

C1

1> $K = H_0(b \| ID_i \| T_{reg})$. 是一个口令保护的 long-term secret，保存在 card 中，S 可以通过数据库的值生成 K。仅破坏 card 而获 password 不可行，但用户可用 card 计算 K.

2> 在更新 N_i 前，验证 A_i 的值是否正确，且更改在本地发生，满足 C2 和 C4

但除了 N_i，还有其它元素存储在 card 中，例如 A_i（若为 $H_0(ID_i \| b \| H_0(PW_i))$，会导致离线类攻击

而若 $A_i = H_0(H_0(ID_i) \oplus H_0(PW_i)) \bmod n_0$），即使敌手已知 ID，它也需做 $\frac{|ID| \cdot |PW|}{n_0}$ 次在线猜测

而这可以用 honey words 做及时检测。（用户误输入口令也可及时检测 C9）.

A_i 称为 "a fuzzy verifier"（未存储于 Server，避免S破坏泄漏 A_i）.

在 card 被破坏时

3> 个可能临时得到SC并获取了 A_i，更改口令并执行 ① 尝试登录 ② 返回 card.

①：个使用 PW_i^* 得到了 A_i，与存储值相同，但 PW 正确的概率才 $\frac{1}{2^{|PW|}}$.

而只有正确的口令才能得到 K，故错误的 K' 会被 honey words 检测到

② 代表 DoS 攻击，由于有阈值限定，个很难成功（U还可以重新行注册）.

4> 采用 $H_0(b \| PW_i)$ 而非 PW_i 或 $h(PW_i)$，其中 b 为 S 未知的 b 随机值，可满足 C3.

5> S中存有（ID_i 和 T_{reg}），当 card 撤销时，只更新 T_{reg} 即可，满足 C6.

6> 可使用 nonce 替代时间戳代表消息的 freshness，满足 C8.

7> 使用基于会话生成的伪实体 CID_i，满足 C11

8> 使用 DH key exchange 实现 C12.

使用大规模实际口令来表明 fuzzy-verifier 有效性；

敌手可以获取SC，并得到 A_i，通过离线方式将候选口令减少为 $\frac{|ID|}{n_0}$.

假设为漫步猜测攻击者，它会先攻击流行口令，使用 guessing entropy (GE) 捕获攻击策略：$G(D) = \sum_{i=1}^{|D|} p_i \cdot i$，即敌手做这些次猜测可得到口令

分析得到当实际数据集大于300万时，$n_0 = 2^8$，有 $GE > 2^{12}$，即在线猜测次数

fuzzy-verifier + honey words 的有效性

事件 Ext：U_i 的 card 被破坏，S 及时检测到.

在V3中，A_i 正确，K 不正确，S 可知（$1-\varepsilon$）的概率 Ext 发生，若允许 login m_0 次

则概率为：$(1-(\frac{1}{n_0})^{m_0})$,

$Pr[Err\text{-}change] = \frac{1}{m_0}$，用户误输入，$Pr[Err\text{-}detect] = \frac{1}{n_0^{m_0}}$. 检测出错误，SC未compt

$Pr[Succ\text{-}Ext] = C' \cdot m_0^{\beta}$，Ext发生且个得到了PW.

上述3个值应达到一个平衡，本文推荐：$n_0 = 2^8$, $m_0 = 10$.

形式化安全分析：CDH，BPR安全模型，ROM，Find-to-Guess。

$\mathcal{A}$可做query：Execute(·)、Send(·)，Test(·)，Reveal(·)。C-3

Corrupt(I,a)：I=U, a=1，输出U的吐。C-2
I=U, a=2，输出SC的安全参数。
I=S, a=1，输出私钥及DB中的值。C-4
（可抽获 known attacks）（C5）

Freshness：①I已accept且生成了sk ②I和其partner未被Reveal。
③至多一种Corrupt(·)指向U。

证明语义安全性（未考虑 forward secrecy）。$G_0 \sim G_8$。

对于每个Game定义事件：①$Succ_n$：$\mathcal{A}$正确猜对了Test-query中的bit c
②$AskPara_n$：$\mathcal{A}$通过询问H_0计算出了k。
③$AskAuth_n$：$\mathcal{A}$计算出了k并且询问H_0或H_2算出C_2或C_3。
④$AskH_n$：$\mathcal{A}$询问H_i（$ID_i||ID_s||Y_i||C_3||k||K$）。

G_0：ROM下的真实攻击，$Adv_P^{ake}(\mathcal{A}) = 2Pr[Succ_0] - 1$

G_1：simulate了各个H_i及H_i（Game 7），并维护Λ_H和Λ_A，并simulate所有instance以便$\mathcal{A}$做各种query，易知：$|Pr[Succ_1] - Pr[Succ_0]| = 0$。

G_2：simulate排除了G_1中不太可能的碰撞（若碰撞，$\mathcal{A}$可用Reveal轻易获胜）
①消息（$(C_1, M_2, CAK_2; CID_2)$，(C_3, C_4)）的碰撞。
②hash输出的碰撞。

由生日悖论：$|Pr[Succ_2] - Pr[Succ_1]| \le \frac{(q_{send}+q_{exe})^2}{2p} + \frac{q_{H_i}}{2^{l_i-1}}$。$l_i$为$l_0 \sim l_3$的min

G_3：simulate排除敌手幸运猜到了C_3, C_4（未询问H_1和H_2），由于C_1和C_2在前面的论证未出现（C_2已排除），故$\mathcal{A}$只能直接猜测C_3, C_4才会发生：

$$|Pr[Succ_3] - Pr[Succ_2]| \le \frac{q_{send}}{2^{l_0}}$$

G_4：simulate排除敌手猜对k的可能：
U在计算C_4，sk_U前，检查是否查询H_0生成过k，若未生成则abort，否则计算C_4，sk_U
S在计算C_4^*，若与C_4相同，检测$\mathcal{A}$是否查询H_0或是否存在相应记录，若不存在则abort
由于C_0, C_2第一次出现，故上述发生只能是$\mathcal{A}$猜到了k。

$$|Pr[Succ_4] - Pr[Succ_3]| \le \frac{q_{send}}{2^{l_0}}$$

G_5：simulate排除$\mathcal{A}$计算得到k，并伪装成了U或S。
U计算k_U和C_3前，检查$\mathcal{A}$是否有询问得k的记录，有则abort 伪装成S
S计算C_4^*后，若$\Lambda_{\mathcal{A}}$中存在H_0的记录或有Λ_{Ψ}中的记录，则abort 伪装成C
$|Pr[Succ_5] - Pr[Succ_4]| \le Pr[AskPara_5]$。（后续假设$\mathcal{A}$正确计算k）

G_6：simulate排除$\mathcal{A}$通过H_1, H_2计算得到C_3, C_4并伪装成U和S
若$\Lambda_{\mathcal{A}}$中存在计算C_3记录，则abort
若$\Lambda_{\mathcal{A}}$中存在请求H_0或计算C_4，或Λ_{Ψ}中有对应记录，则abort。
$|Pr[Succ_6] - Pr[Succ_5]| \le Pr[AskAuth_6]$
$|Pr[AskPara_6] - Pr[AskPara_5]| \le Pr[AskAuth_6]$

G_7：使用private oracle：H_0'产生$(ID_i||ID_s||C_3||C_4)$。$\mathcal{A}$未知
则C_3, C_4，sk_U、sk_S与k，k_U和k_S无关，除非$AskH_7$发生，否则G_6, G_7不区分
$|Pr[Succ_7] - Pr[Succ_6]| \le Pr[AskH_7]$，$|Pr[AskPara_7] - Pr[AskPara_6]| \le Pr[AskH_7]$
$|Pr[AskAuth_7] - Pr[AskAuth_6]| \le Pr[AskH_7]$
易知，$Pr[Succ_7] = \frac{1}{2}$

R(U)代表U收到的(C_2, C_3)的集合，R(S)代表S收到的C_4的集合。
$\mathcal{A}$成功算出k，存在两种可能 Corrupt(I,1)；Corrupt(I,2)
$|Pr[AskPara_7 With Corr_1]|$为R(U)中存在$\mathcal{A}$生成的$(C_2, C_3)$加R(S)中存在$\mathcal{A}$生成的$C_4$的不同概率和 $\le C' \cdot q_{send}^{s'}$
$Pr[AskPara_7 With Corr_2] \le \frac{q_{send}}{2^{l_0}}$

G_8：simulate，使用 random self-reducibility of DH problem.
给定CDH实例(A,B)
U_1：$\alpha \in_R Z_p^*$，$C_3 = A^{\alpha} \bmod p$，…，Λ_A添加元素(C_1, α)。
S_2：$\alpha \in_R Z_p^*$，$C_3 = B^{\beta} \bmod p$，…，Λ_B添加元素(C_2, β)。
$Pr[AskH_7] = Pr[AskH_8]$，→ $\mathcal{A}$询问了$H_i(ID_i||ID_s||Y_i||C_2||K)$ CDH(C_1, C_2)
则在Λ_A和Λ_B中找 $C_1 = A^{\alpha}$，$C_2 = B^{\beta}$，$CDH(C_1, C_2) = CDH(A^{\alpha}, B^{\beta}) = CDH(A,B)^{\alpha\beta}$
$|Pr[AskH_8]| \le q_h \cdot Adv_P^{CDH}(t')$　$t' \le t + (q_{send} + q_{exe} + 1) \cdot T_e$。得证

(3)