Password Hashing and Preprocessing

Pooya Farshim

Stefano Tessaro

Passwords

- Still one of the most prevalent forms of authentication
- Use H(pw) in place of pw
- Use H(pw) as a secret key (e.g., to encrypt data)

The Setting

- Attackers attempt to “crack” multiple passwords
- May well use preprocessing (e.g. Rainbow Tables) to speed things up
- Typically the hash function is assumed to be secure (= RO)
- Attackers exploit the low entropy of human-generated passwords

Conventional Wisdom

$\mathsf{sa}_i, \mathsf{H}(\mathsf{sa}_i, \mathsf{pw}_i)$

"Salting Defeats Preprocessing"

Distinct (known) salts $\Longrightarrow$ separate preprocessing effort

Unpredictable salts $\Longrightarrow$ defeat preprocessing effort

Related Works

- Long line of works on preprocessing: [GT00,Unr07,DTT10,DGK17,CDGS18,CDG18,...]
- Bellare, Ristenpart, Tessaro [BRT12]: Highlight the need for multi-instance security in password-based crypto.

Two Settings

Preprocessing

- $\sigma \leftarrow A_0(\mathsf{H})$
- Run G with $A_1^{\mathsf{H}}(\sigma)$

Two Settings

Preprocessing

- $\sigma \leftarrow A_0(\mathsf{H})$
- Run G with $A_1^{\mathsf{H}}(\sigma)$

Multi-Instance

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}\big(\mathsf{H}(\mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{pw}_m)\big)$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

The Multi-Instance Perspective

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}(\mathsf{H}(\mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{pw}_m))$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

- Effort to compromise *a* single instance may be within reach
- Want: effort to compromise m instances to scale with m

The Multi-Instance Perspective

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}(\mathsf{H}(\mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{pw}_m))$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

- Effort to compromise *a* single instance may be within reach
- Want: effort to compromise m instances to scale with m
- Not without salting:

$$\mathbf{Adv}^{\mathrm{rec}}_{[N]^m,\perp,\mathsf{RO}}(T) \geq \left(\frac{T}{N}\right)^m$$

The Multi-Instance Perspective

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}\big(\mathsf{H}(\mathsf{sa}_1, \mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{sa}_n, \mathsf{pw}_m)\big)$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

The Multi-Instance Perspective

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}\big(\mathsf{H}(\mathsf{sa}_1, \mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{sa}_n, \mathsf{pw}_m)\big)$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

- With sufficiently large salts:

$$\mathbf{Adv}^{\mathrm{rec}}_{[N]^m,[K],\mathsf{RO}}(T) \geq \left(\frac{T}{m \cdot N}\right)^m$$

Our Goal

Understand the security of password hashing
in the presence of multiple instances and preprocessing

The Setting

Preprocessing

- $\sigma \leftarrow A_1(\mathsf{H})$
- Run G with $A_1^{\mathsf{H}}(\sigma)$

Multi-Instance

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}\big(\mathsf{H}(\mathsf{sa}_1, \mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{sa}_m, \mathsf{pw}_m)\big)$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

The Setting

Preprocessing

- $\sigma \leftarrow A_1(\mathsf{H})$
- Run G with $A_1^{\mathsf{H}}(\sigma)$

Multi-Instance

- $(\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m) \leftarrow A^{\mathsf{H}}\big(\mathsf{H}(\mathsf{sa}_1, \mathsf{pw}_1), \ldots, \mathsf{H}(\mathsf{sa}_m, \mathsf{pw}_m)\big)$
- Return $(\mathsf{pw}_1, \ldots, \mathsf{pw}_m) = (\mathsf{pw}'_1, \ldots, \mathsf{pw}'_m)$

Combine!

We Consider

No Salts, Distinct Salts, & Random Salts

We Consider

No Salts, Distinct Salts, & Random Salts

Historical, Amplification

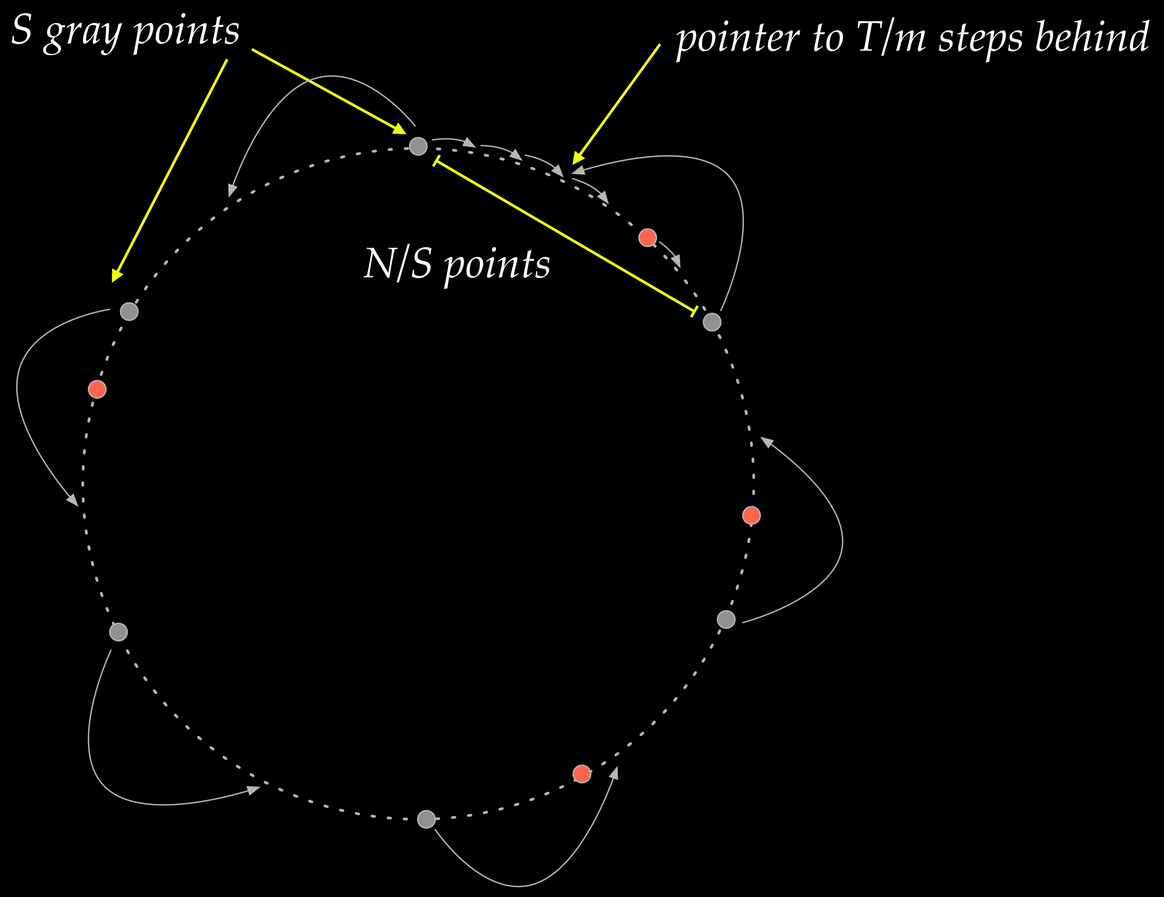
We Consider

No Salts, Distinct Salts, & Random Salts

Historical, Amplification

Practice

Multi-Instance Hellman



$$\Pr\left[\textbf{All reds within } T/m \textbf{ of a gray}\right] \approx \left(\frac{T/m}{N/S}\right)^m = \left(\frac{ST}{mN}\right)^m$$

Password Unguessability [BRT12]

Game $\textsc{Guess}_{\mathcal{P}}^{\mathcal{A}}$:	Proc. $\textsc{Test}(i, pw)$:	Proc. $\textsc{Cor}(i)$:
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$	$\mathsf{win}_i \leftarrow (pw = \mathbf{pw}[i])$	$\mathsf{win}_i \leftarrow \mathsf{true}$
$y \twoheadleftarrow \mathcal{A}^{\textsc{Test},\textsc{Cor}}(z)$	return win_i	return $\mathbf{pw}[i]$
return $\bigwedge_{i=1}^{m} \mathsf{win}_i$		

$$\mathbf{Adv}_{\mathcal{P}}^{\text{guess}}(T, c) := \max_{\mathcal{A}} \Pr\left[\textsc{Guess}_{\mathcal{P}}^{\mathcal{A}}\right]$$

The Basic Measure

Game $\textsc{Guess}^{\mathcal{A}}_{\mathcal{P}}$:	Proc. $\textsc{Test}(i, pw)$:	Proc. $\textsc{Cor}(i)$:
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$	$\mathsf{win}_i \leftarrow (pw = \mathbf{pw}[i])$	$\mathsf{win}_i \leftarrow \mathsf{true}$
$y \twoheadleftarrow \mathcal{A}^{\textsc{Test},\textsc{Cor}}(z)$	return win_i	return $\mathbf{pw}[i]$
return $\bigwedge_{i=1}^{m} \mathsf{win}_i$		

$$\mathbf{Adv}^{\mathrm{guess}}_{\mathcal{P}}(m, 0) = 2^{-\tilde{\mathbf{H}}_{\infty}(\mathcal{P}|\mathcal{Z})}$$

The Basic Measure

Game $\textsc{Guess}^{\mathcal{A}}_{\mathcal{P}}$:	Proc. $\textsc{Test}(i, pw)$:	Proc. $\textsc{Cor}(i)$:
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$ $y \twoheadleftarrow \mathcal{A}^{\textsc{Test},\textsc{Cor}}(z)$ return $\bigwedge_{i=1}^{m} \mathsf{win}_i$	$\mathsf{win}_i \leftarrow (pw = \mathbf{pw}[i])$ return win_i	$\mathsf{win}_i \leftarrow \mathsf{true}$ return $\mathbf{pw}[i]$

$$\mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(m - c, c)$$

A Simple Argument

Theorem 2 (Unguessability). *For any m-sampler $\mathcal{P}$ and any $T, c \in \mathbb{N}$,*

$$\mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(T, c) \leq \binom{T}{m-c} \cdot \mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(m-c, c) \ .$$

A Simple Argument

Theorem 2 (Unguessability). *For any m-sampler $\mathcal{P}$ and any $T, c \in \mathbb{N}$,*

$$\mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(T, c) \leq \binom{T}{m-c} \cdot \mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(m-c, c) \ .$$

Proof:

- Guess the (m — c) out of T queries to TEST that will return T. Relay these to own TEST.
- Answer the rest of TEST queries with F.
- Relay all COR queries to own COR.

Resolves Open Question in [BRT12]

DISCUSSION. We note that although the bound in Theorem 3.2 depends only on q_t, q_c, we are not claiming that it holds for any (q_t, q_c)-guessing adversary, but only for a $(\mathbf{q}, q_c)$-guessing adversary with $q_t = \mathbf{q}[1] + \cdots + \mathbf{q}[m]$. The same is true for the better bound we just quoted for the $q_c = 0$ case. An example from [27] shows that our bound of $(q_t/m2^{\mu})^m$ for the $q_c = 0$ case in fact does not hold for arbitrary (q_t, q_c)-guessing adversaries. For $q_t = 11$, $m = 2$ and $\mathcal{P}$ which picks random, 3-bit passwords, they present an attack with advantage higher than $(q_t/m2^{\mu})^m$. Obtaining a (good) bound for (q_t, q_c)-guessing adversaries is an interesting open question.

- BRT considered *a priori* bounds T[i] for *each* index i.
- We remove this restriction:

  Adversary can adapt T[i] as it makes progress

So far...

We looked at
unguessability of passwords in terms of the basic measure.

So far...

We looked at
unguessability of passwords in terms of the basic measure.

Next: hashed salted passwords

Unrecoverability in AI-RO

Game $\text{AI-REC}^{\mathcal{A}_0,\mathcal{A}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}}$:

$\mathsf{H} \twoheadleftarrow \mathrm{Fun}(D, R)$
$\sigma \twoheadleftarrow \mathcal{A}_0(\mathsf{H})$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i, j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i, j] \twoheadleftarrow \mathsf{Gen}(i, j)$
  $\mathbf{k}[i, j] \leftarrow \mathsf{KD}^{\mathsf{H}}(\mathbf{pw}[i], \mathbf{sa}[i, j])$
$\mathbf{pw}' \twoheadleftarrow \mathcal{A}_1^{\mathsf{H},\text{COR}}(\mathbf{sa}, \mathbf{k}, \sigma, z)$
return $(\mathbf{pw}' = \mathbf{pw})$

Proc. $\text{COR}(i)$:

return $\mathbf{pw}[i]$

Unrecoverability in AI-RO

Game $\text{AI-REC}^{\mathcal{A}_0,\mathcal{A}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}}$:

$\mathsf{H} \twoheadleftarrow \mathrm{Fun}(D,R)$
$\sigma \twoheadleftarrow \mathcal{A}_0(\mathsf{H})$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i,j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i,j] \twoheadleftarrow \mathsf{Gen}(i,j)$
  $\mathbf{k}[i,j] \leftarrow \mathsf{KD}^{\mathsf{H}}(\mathbf{pw}[i], \mathbf{sa}[i,j])$
$\mathbf{pw}' \twoheadleftarrow \mathcal{A}_1^{\mathsf{H},\text{COR}}(\mathbf{sa}, \mathbf{k}, \sigma, z)$
return $(\mathbf{pw}' = \mathbf{pw})$

Proc. $\text{COR}(i)$:
return $\mathbf{pw}[i]$

Game $\text{BF-REC}^{\mathcal{A}_0,\mathcal{A}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}}$:

$\mathsf{H} \twoheadleftarrow \mathrm{Fun}(D,R)$
$(\sigma, L) \twoheadleftarrow \mathcal{A}_0()$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i,j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i,j] \twoheadleftarrow \mathsf{Gen}(i,j)$
  $\mathbf{k}[i,j] \leftarrow \mathsf{KD}^{\mathsf{H}[L]}(\mathbf{pw}[i], \mathbf{sa}[i,j])$
$\mathbf{pw}' \twoheadleftarrow \mathcal{A}_1^{\mathsf{H}[L],\text{COR}}(\mathbf{sa}, \mathbf{k}, \sigma, z)$
return $(\mathbf{pw}' = \mathbf{pw})$

Proc. $\text{COR}(i)$:
return $\mathbf{pw}[i]$

Unrecoverability in AI-RO

[CDGS18]

Game $\text{AI-REC}^{\mathcal{A}_0,\mathcal{A}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}}$:

$\mathsf{H} \twoheadleftarrow \text{Fun}(D,R)$
$\sigma \twoheadleftarrow \mathcal{A}_0(\mathsf{H})$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i,j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i,j] \twoheadleftarrow \mathsf{Gen}(i,j)$
  $\mathbf{k}[i,j] \leftarrow \mathsf{KD}^{\mathsf{H}}(\mathbf{pw}[i], \mathbf{sa}[i,j])$
$\mathbf{pw}' \twoheadleftarrow \mathcal{A}_1^{\mathsf{H},\text{COR}}(\mathbf{sa}, \mathbf{k}, \sigma, z)$
return $(\mathbf{pw}' = \mathbf{pw})$

Proc. $\text{COR}(i)$:
return $\mathbf{pw}[i]$

Game $\text{BF-REC}^{\mathcal{A}_0,\mathcal{A}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}}$:

$\mathsf{H} \twoheadleftarrow \text{Fun}(D,R)$
$(\sigma, L) \twoheadleftarrow \mathcal{A}_0()$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i,j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i,j] \twoheadleftarrow \mathsf{Gen}(i,j)$
  $\mathbf{k}[i,j] \leftarrow \mathsf{KD}^{\mathsf{H}[L]}(\mathbf{pw}[i], \mathbf{sa}[i,j])$
$\mathbf{pw}' \twoheadleftarrow \mathcal{A}_1^{\mathsf{H}[L],\text{COR}}(\mathbf{sa}, \mathbf{k}, \sigma, z)$
return $(\mathbf{pw}' = \mathbf{pw})$

Proc. $\text{COR}(i)$:
return $\mathbf{pw}[i]$

BF-Rec Bound (Uniform Salts)

Theorem 7. *Let* $\mathsf{KD}^{\mathsf{H}}(pw, sa) := \mathsf{H}(pw|sa)$ *for random oracle* H. *Then for any* m*-sampler* $\mathcal{P}$*, any salt generator* $\mathsf{Gen} := [K]$ *that outputs uniform salts in a set of size* K*, and any* $\ell, P, T, c \in \mathbb{N}$*,*

$$\mathbf{Adv}^{\text{bf-rec}}_{\mathcal{P},\ell,[K],\mathsf{H}}(P,T,c) \leq \left(\binom{T}{m-c} + \frac{m\ell}{K}\binom{T+P}{m-c}\right) \cdot \mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(m-c,c) + \frac{m^2\ell^2}{K} \ .$$

Other Bounds (See Paper)

Theorem 6. *Let* $\mathsf{KD}^{\mathsf{H}}(pw, sa) := \mathsf{H}(pw|sa)$ *for random oracle* H. *Then for any* m*-sampler* $\mathcal{P}$*, any salt generator* Gen*, and any* $\ell, P, T, c \in \mathbb{N}$*,*

$$\mathbf{Adv}^{\text{bf-rec}}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{H}}(P, T, c) \leq \mathbf{Adv}^{\text{sa-guess}}_{\mathcal{P},\ell,\mathsf{Gen}}(T + P, c) \ .$$

Theorem 4. *For any* m*-sampler* $\mathcal{P}$ *and any* $\ell, T \in \mathbb{N}$*,*

$$\mathbf{Adv}^{\text{sa-guess}}_{\mathcal{P},\ell,\perp}(T, 0) \leq T^m \cdot \mathbf{Adv}^{\text{guess}}_{\mathcal{P}}(m, 0) \ .$$

Our Main Bounds

	No salts	Known distinct salts	Uniform salts
$S = 0$	$\left(\frac{6T}{N}\right)^m$	$\left(\frac{6\mathrm{e}T}{mN}\right)^m$	$\left(1 + \frac{m\ell}{K}\right) \cdot \left(\frac{6\mathrm{e}T}{mN}\right)^m$
"Large" $S \geq 3m$	$\left(\frac{6ST}{mN}\right)^m$	$\left(\frac{6\mathrm{e}ST}{m^2N}\right)^m$	$\left(\frac{2\mathrm{e}T}{mN}\right)^m + \frac{m\ell}{K} \cdot \left(\frac{6\mathrm{e}ST}{m^2N}\right)^m$

Composable AI-KDF Security (Following BRT)

Game AI-KDF-REAL$^{\mathcal{D}_0,\mathcal{D}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}}$:

$\mathsf{H} \twoheadleftarrow \mathrm{Fun}(D,R)$
$\sigma \twoheadleftarrow \mathcal{D}_0(\mathsf{H})$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i,j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i,j] \twoheadleftarrow \mathsf{Gen}(i,j)$
  $\mathbf{k}[i,j] \leftarrow \mathsf{KD}^{\mathsf{H}}(\mathbf{pw}[i], \mathbf{sa}[i,j])$
$b' \twoheadleftarrow \mathcal{D}_1^{\text{PRIM}}(\mathbf{pw}, \mathbf{sa}, \mathbf{k}, z, \sigma)$
return b'

Proc. PRIM(w):

return $\mathsf{H}(w)$

Game BF/AI-KDF-IDEAL$^{\mathcal{D}_1}_{\mathcal{P},\ell,\mathsf{Gen},\mathcal{S}_0,\mathcal{S}_1}$:

$(\sigma, st) \twoheadleftarrow \mathcal{S}_0()$
$(\mathbf{pw}, z) \twoheadleftarrow \mathcal{P}$
for $(i,j) \in [m] \times [\ell]$ do
  $\mathbf{sa}[i,j] \twoheadleftarrow \mathsf{Gen}(i,j)$
  $\mathbf{k}[i,j] \twoheadleftarrow \{0,1\}^k$
$b' \twoheadleftarrow \mathcal{D}_1^{\text{PRIM}}(\mathbf{pw}, \mathbf{sa}, \mathbf{k}, z, \sigma)$
return b'

Proc. PRIM(w):

$(y, st) \twoheadleftarrow \mathcal{S}_1^{\text{TEST}}(w; st)$
return y

Proc. TEST(pw, sa):

$S \leftarrow \{i \in [m] : \exists j \in [\ell] \text{ st. } (\mathbf{pw}[i], \mathbf{sa}[i,j]) = (pw, sa)\}$
return $\mathbf{k}[S]$

Theorem 13. AI-KDF security composes in the presence of *aux info* and multiple instances:

$$\mathbf{Adv}^{\text{ai-multi}}_{\mathrm{G},\mathcal{P},\mathsf{Gen},\mathsf{KD}}(\mathcal{A}_0,\mathcal{A}_1) \le 2 \cdot \mathbf{Adv}^{\text{ai-kdf}}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD},\mathcal{S}_0,\mathcal{S}_1}(\mathcal{D}_0,\mathcal{D}_1) + \\ + 2 \cdot \mathbf{Adv}^{\text{sa-guess}}_{\mathcal{P},\ell,\mathsf{Gen}}(T,c) + m \cdot \mathbf{Adv}^{\text{single}}_{\mathrm{G}}(\mathcal{B}) \ .$$

AI-KDF Security of Iteration

$$\mathsf{KD}_r^{\mathsf{H}}(pw, sa) := \underbrace{\mathsf{H} \circ \cdots \circ \mathsf{H} \circ \mathsf{H}}_{r}(pw|sa)$$

BF-KDM Security:

$$\mathbf{Adv}^{\text{bf-kdf}}_{\mathcal{P},\ell,\mathsf{Gen},\mathsf{KD}_r,\mathcal{S}_0,\mathcal{S}_1}(\mathcal{D}_0, \mathcal{D}_1) \leq \frac{(r+1)m\ell T}{N} + 3 \cdot \left(\frac{rm\ell(rm\ell + P)}{N} + \frac{m\ell P}{K} + \frac{m^2\ell^2}{K} \right)$$

(Next: use [CDGS18] and optimize P to get an AI-KDF bound.)

Take-Away Message

	No salts	Known distinct salts	Uniform salts
$S = 0$	$\left(\frac{6T}{N}\right)^m$	$\left(\frac{6\mathrm{e}T}{mN}\right)^m$	$\left(1 + \frac{m\ell}{K}\right) \cdot \left(\frac{6\mathrm{e}T}{mN}\right)^m$
"Large" $S \geq 3m$	$\left(\frac{6ST}{mN}\right)^m$	$\left(\frac{6\mathrm{e}ST}{m^2N}\right)^m$	$\left(\frac{2\mathrm{e}T}{mN}\right)^m + \frac{m\ell}{K} \cdot \left(\frac{6\mathrm{e}ST}{m^2N}\right)^m$

Plus: Composable AI-KDF security of iterated hashing.

Thanks.