Decisional second-preimage resistance

When does SPR imply PRE?

Daniel J. Bernstein, Andreas Hülsing

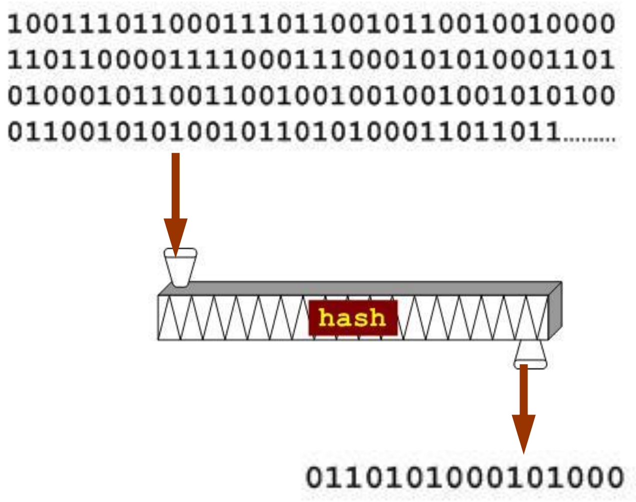
Motivation

This work

- answers a long standing subtle question about the relation of hash function properties
- provides a tool that enables tight security proofs for hash-based signatures

Cryptographic hash functions

- Efficient function
  $\mathrm{h}\colon \{0,1\}^{n} \times \{0,1\}^{l(n)} \rightarrow \{0,1\}^{n}$
- We write $\mathrm{h}(\mathrm{k}, \mathrm{x}) = h_k(x)$
- Key k in this case is public information. Think of function description.



Collision resistance

Success probability of an adversary A against *collision resistance (CR) of h is defined as*

$$\mathrm{Succ}_h^{CR}(A) = \Pr[k \leftarrow_R \{0,1\}^n, (x_1, x_2) \leftarrow A(k): h_k(x_1) = h_k(x_2) \wedge (x_1 \neq x_2)]$$

Second-preimage resistance (SPR)

Success probability of an adversary A against *second-preimage resistance (SPR) of h is defined as*

$$\mathrm{Succ}_h^{SPR}(A) = \Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^{l(n)},$$

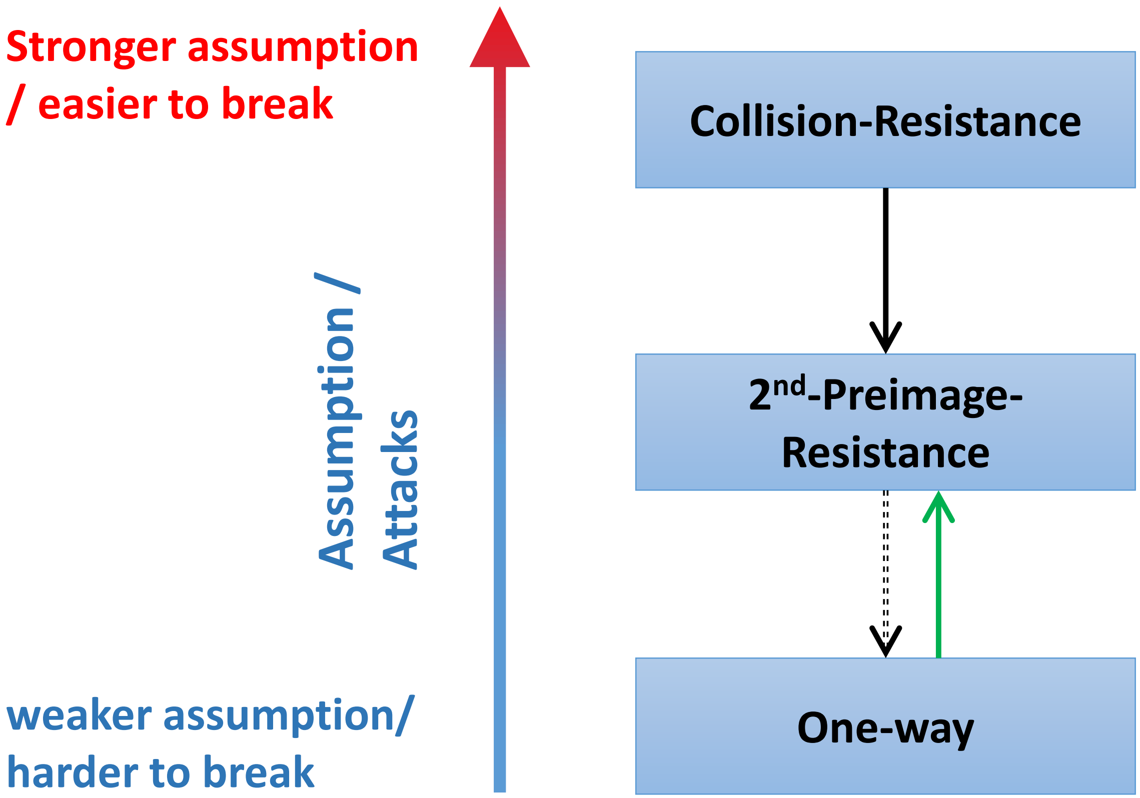
$$x' \leftarrow A(k, x) : h_k(x) = h_k(x') \wedge (x \neq x')]$$

Security properties: Preimage resistance / One-wayness

Success probability of an adversary A against *preimage resistance (PRE) of h is defined as*

$$\mathrm{Succ}_h^{PRE}(A) = \Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^{l(n)},$$
$$y \leftarrow h_k(x), x' \leftarrow A(k, y) : h_k(x') = y]$$

Relations


Stronger assumption
/ easier to break
Assumption /
Attacks
weaker assumption/
harder to break
Collision-Resistance
2nd-Preimage-
Resistance
One-way

CR implies SPR?

Reduction $B_{CR}^{A_{SPR}}(k)$:

1. $x \leftarrow_R \{0,1\}^{l(n)}$
2. $x' \leftarrow A_{SPR}(k, x)$
3. Return (x, x')

$$\mathrm{Succ}_h^{CR}\left(B_{CR}^{A_{SPR}}\right) = \mathrm{Succ}_h^{SPR}(A_{SPR})$$

SPR implies PRE?

Reduction $B_{SPR}^{A_{PRE}}(k, x)$:

1. $y = h_k(x)$
2. $x' \leftarrow A_{PRE}(k, y)$
3. Return x'

$\mathrm{Succ}_h^{SPR}\left(B_{SPR}^{A_{PRE}}\right) \geq^? 0.5 \cdot \mathrm{Succ}_h^{PRE}(A_{PRE})$

Where is the problem?

Positive result

Rogaway-Shrimpton show that for $l(n)$ much bigger then n we are fine

Negative result

The identity function demonstrates that SPR cannot generally imply PRE.

The gap

Functions with $l(n) \approx n$
(especially length preserving)

Exactly the ones we use
in hash-based OTS

Are we doomed?

The general case

- SHA-X $\neq$ identity function
- SHA-X $\approx$ random function

Fooling the reduction

- Reductions have to work **for all** A!

- $A(k, y)$:
  1. Compute $X = f_k^{-1}(y)$
  2. If $|X| > 1$, abort
  3. Else $X = \{x\}$, return x

For $f_k: \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ random,

$$\mathrm{Succ}_h^{PRE}(A) = \frac{1}{e}$$

$$\mathrm{Succ}_h^{SPR}\left(B_{SPR}^{A_{PRE}}\right) = 0$$

Decisional second-preimage resistance to the rescue!

- $P_k(x) = \begin{cases} 0, \text{ if } \left|f_k^{-1}(f_k(x))\right| = 1 \\ 1, \text{ otherwise} \end{cases}$
- Can salvage reduction $B_{SPR}^{A_{PRE}}$ if
  1. $\mathrm{SPprob}(\mathrm{f_k}) = \Pr[P_k(x) = 1 \mid x \leftarrow_R \{0,1\}^n]$ is non-negligible , and
  2. it is hard to reliably determine $P_k(x)$

We show that $\mathrm{SPprob}(\mathrm{f_k}) \approx \left(1 - \frac{1}{e}\right)$ for the overwhelming majority of all functions. E.g., for a random 256bit hash $\mathrm{f_k}$

$$\Pr[\mathrm{SPprob}(\mathrm{f_k}) < 0.6] \approx 2^{-2^{239}}$$

DSPR = "It is hard to reliably determine $P_k(x)$"

$$\mathrm{Adv}_f^{DSPR}(A) = \max\{0, \Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^n, b \leftarrow A(k,x) : P_k(x) = b] - \mathrm{SPprob}(\mathrm{f_k})\}$$

Some intuition about DSPR

- $[\mathrm{SPprob}(\mathrm{f_k}) \approx 1] \Rightarrow \left[\mathrm{Adv}_f^{DSPR}(A) \approx 0\right]$
- If $\mathrm{f_k}$ is strongly compressing,
  it is information-theoretically DSPR.

$$\mathrm{Adv}_f^{DSPR}(A) = \max\{0,$$
$$\Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^n, b \leftarrow A(k,x): P_k(x) = b]$$
$$-\mathrm{SPprob}(\mathrm{f_k})\}$$

Some intuition about DSPR

- Given SPR adversary A_{SPR} define:
- $B_{DSPR}^{A_{SPR}}(k, x)$:

1. If $A_{SPR}(k, x)$ succeeds, return 1
2. Return 0.

- Requires $\mathrm{Succ}_f^{SPR}(A_{SPR}) > 2 \cdot \mathrm{SPprob}(\mathrm{f_k}) - 1$ for non-zero advantage!

$$\mathrm{Adv}_f^{DSPR}(A) = \max\{0,$$
$$\Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^n, b \leftarrow A(k, x): P_k(x) = b]$$
$$-\mathrm{SPprob}(\mathrm{f_k})\}$$

DSPR at work

Reduction $B_{SPR}^{A_{PRE}}(k, x)$:

1. $y = h_k(x)$
2. $x' \leftarrow A_{PRE}(k, y)$
3. Return x'

Reduction $C_{DSPR}^{A_{PRE}}(k, x)$:

1. $y = h_k(x)$
2. $x' \leftarrow A_{PRE}(k, y)$
3. Return 1 if $x' \neq x$
4. Return 0

We show

$$\mathrm{Succ}_f^{PRE}(A_{PRE})$$
$$\leq \mathrm{Adv}_f^{DSPR}\left(C_{DSPR}^{A_{PRE}}\right) +$$
$$3 \cdot \mathrm{Succ}_f^{SPR}\left(B_{SPR}^{A_{PRE}}\right)$$

Application to hash-based signatures

- Interactive Game T-OpenPRE:

1. Generate T pairs $(k_i, y_i) = \left(k_i, f_k(x_i)\right)$,
$$x_i \leftarrow_R \{0,1\}^n$$
2. Give pairs to A and allow A to ask for up to $T - 1$ of the x_i
3. Output 1 if $(j, x) \leftarrow A(\,)$ is a preimage ($f_{k_j}(x) = y_j$) for "unopened" image y_j

Variants of this naturally arise in security proof of WOTS, and L-OTS

Pre ⇒ T-OpenPRE is non-tight!

Given $A_{T-OpenPRE}$

build $B(k, y)$:

1. Play T-OpenPRE game but replace random pair (k_c, y_c) by challenge (k, y)
2. If A asks to open position c, abort
3. If A returns (i, x), output x

Reduction loss of 1/T!

Multi-target DSPR

Definition 31 ($\boldsymbol{T}$-DSPR). *Let T be a positive integer. Let $\mathcal{A}$ be an algorithm with output in $\{1, \ldots, T\} \times \{0, 1\}$. The advantage of $\mathcal{A}$ against the T-target decisional second-preimage resistance of a keyed hash function* H *is*

$$\mathrm{Adv}_{\mathrm{H}}^{T\text{-DSPR}}(\mathcal{A}) \stackrel{def}{=} \max\{0, q - p\}$$

where

$$\begin{aligned} q = \mathsf{Pr}\big[&(x_1, k_1, \ldots, x_T, k_T) \leftarrow_R (\mathcal{X} \times \mathcal{K})^T; \\ &(j, b) \leftarrow \mathcal{A}(x_1, k_1, \ldots, x_T, k_T) : P_{k_j}(x_j) = b\big]; \\ p = \mathsf{Pr}\big[&(x_1, k_1, \ldots, x_T, k_T) \leftarrow_R (\mathcal{X} \times \mathcal{K})^T; \\ &(j, b) \leftarrow \mathcal{A}(x_1, k_1, \ldots, x_T, k_T) : P_{k_j}(x_j) = 1\big]; \end{aligned}$$

and $P_{k_j} = \mathrm{SPexists}(\mathrm{H}_{k_j})$.

T-DSPR + T-SPR ⇒ T-OpenPRE, tightly!

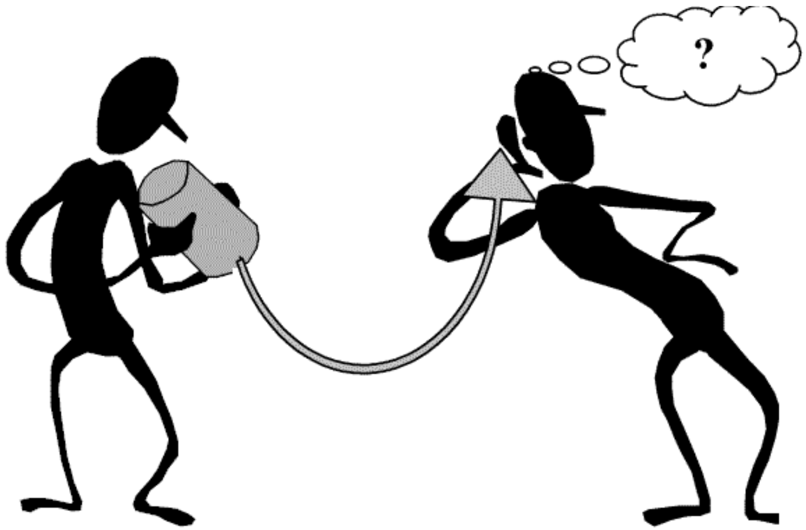
- Use T-target versions of $B_{SPR}^{A_{PRE}}$, and $C_{DSPR}^{A_{PRE}}$
- Can replace **all** pairs by challenges
  - We do know a preimage
- If $A_{T-OpenPRE}$ always returns known image, $C_{T-DSPR}^{A_{T-OpenPRE}}$ will have advantage in breaking T-DSPR
- Else, $B_{T-SPR}^{A_{T-OpenPRE}}$ succeeds with high probability

More in paper

- DSPR is quantum-hard for random functions
- Detailed analysis of SPprob
- Full proofs

- See “The SPHINCS+ Signature Framework“ (CCS‘19) for application to SPHINCS+ and other hash-based signatures.

Questions?



Paper(s) available at
https://sphincs.org/resources.html