Smooth Projective Hash Function and Its Applications

Rongmao Chen
University of Wollongong

November 21, 2014

Literature

Ronald Cramer and Victor Shoup.
Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption.
In *EUROCRYPT*, pages 13–25, 2002.

Ronald Cramer and Victor Shoup.
A practical public key cryptosystems provably secure against adaptive chosen ciphertext attack.
In *CRYPTO*, pages 13–25, 1998.

Shai Halevi and Yael Tauman Kalai.
Smooth Projective Hashing and Two-Message Oblivious Transfer.
In *Journal of Cryptology* , pages 158-193, 2012.

Rosario Gennaro and Yehuda Lindell.
A Framework for Password-Based Authenticated Key Exchange.
In *EUROCRYPT*, pages 524-543, 2003.

Outline

1. Part I: A Case Study
2. Part II: Smooth Projective Hash Function
3. Part III: Applications of SPHF

Part I: A Case Study-ECP Problem

ECP-Problem

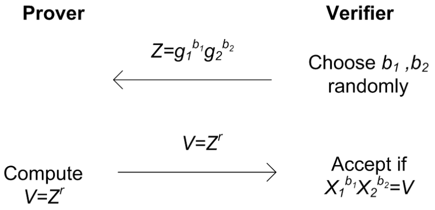
ECP-Problem

Problem: Given g_1, g_2, X_1, X_2, the prover wants to prove to the verifier that there is an r where $X_1 = g_1^r$, $X_2 = g_2^r$ without leaking the value of r to the verifier.

ECP-Problem

Problem: Given g_1, g_2, X_1, X_2, the prover wants to prove to the verifier that there is an r where $X_1 = g_1^r$, $X_2 = g_2^r$ without leaking the value of r to the verifier.

Solution:



Case Study–Exponentiations Consistency Proving (ECP)

ECP-Problem

Problem: Given g_1, g_2, X_1, X_2, the prover wants to prove to the verifier that there is an r where $X_1 = g_1^r$, $X_2 = g_2^r$ without leaking the value of r to the verifier.

Solution:

Prover		**Verifier**
	$\xleftarrow{Z=g_1^{b_1}g_2^{b_2}}$	Choose b_1, b_2 randomly
Compute $V=Z^r$	$\xrightarrow{V=Z^r}$	Accept if $X_1^{b_1}X_2^{b_2}=V$

Correctness: if $X_1 = g_1^r, X_2 = g_2^r$, then

$$X_1^{b_1} X_2^{b_2} = g_1^{rb_1} g_2^{rb_2} = (g_1^{b_1} g_2^{b_2})^r = Z^r$$

Case Study–ECP Problem (cont'd)

Soundness? What if $X_1 = g_1^{r_1}, X_2 = g_2^{r_2}$, where $r_1 \neq r_2$?
(Denote $\log = log_{g_1}$ and suppose that $\log(g_2) = w$)

Case Study–ECP Problem (cont'd)

Soundness? What if $X_1 = g_1^{r_1}, X_2 = g_2^{r_2}$, where $r_1 \neq r_2$?
(Denote $\log = log_{g_1}$ and suppose that $\log(g_2) = w$)

Note that $Z = g_1^{b_1} g_2^{b_2} = g_1^{b_1 + wb_2}$ which constraints (b_1, b_2) to satisfy

$$b_1 + wb_2 = \log(Z) \tag{1}$$

Soundness? What if $X_1 = g_1^{r_1}, X_2 = g_2^{r_2}$, where $r_1 \neq r_2$?
(Denote $\log = log_{g_1}$ and suppose that $\log(g_2) = w$)

Note that $Z = g_1^{b_1} g_2^{b_2} = g_1^{b_1 + wb_2}$ which constraints (b_1, b_2) to satisfy

$$b_1 + wb_2 = \log(Z) \tag{1}$$

If $X_1 = g_1^{r_1}, X_2 = g_2^{r_2}$, where $r_1 \neq r_2$, then

$$X_1^{b_1} X_2^{b_2} = g_1^{r_1 b_1} g_2^{r_2 b_2} = g_1^{r_1 b_1 + r_2 w b_2}.$$

So, for any $h \in \mathbb{G}$, we have $X_1^{b_1} X_2^{b_2} = h$ iff

$$r_1 b_1 + r_2 w b_2 = \log(h) \tag{2}$$

Equations (1), (2) are linearly independent regarding b_1, b_2. Hence,

$$\Pr[X_1^{b_1} X_2^{b_2} = h] = 1/\mathbb{G}$$

The distribution of $X_1^{b_1} X_2^{b_2}$ is uniform in $\mathbb{G}$.

Case Study–ECP Problem

Several Observations from ECP

1. **Designated Verifier.** Only the verifier who has the trapdoor key (b_1, b_2) can do the verification.
2. **Correctness Assurance.** The proof can be computed in two different ways by the prover (Z^r) and the verifier $(X_1^{b_1} X_2^{b_2})$ respectively.
3. **Soundness Assurance.** The prover can only convince the verifier with negligible probability if the statement is false.

Case Study–ECP Problem

Several Observations from ECP

1. **Designated Verifier.** Only the verifier who has the trapdoor key (b_1, b_2) can do the verification.
2. **Correctness Assurance.** The proof can be computed in two different ways by the prover (Z^r) and the verifier $(X_1^{b_1} X_2^{b_2})$ respectively.
3. **Soundness Assurance.** The prover can only convince the verifier with negligible probability if the statement is false.

Designated Verifier Non-Interactive Zero-Knowledge Proof, where the language is as follow,

$$L_{DDH} = \{(u_1, u_2) : \exists r \in Z_p \ s.t. u_1 = g_1^r, u_2 = g_2^r\},$$

where $g_1, g_2 \in \mathbb{G}, \sharp(\mathbb{G}) = p$.

Part II: Smooth Projective Hash Function

Definition of SPHF

Roughly speaking, the definition of SPHF requires the existence of a domain X and an underlying NP language $L \subseteq X$. And SPHF consists of a keyed hash pair $\mathbf{Hash}_{hk} : X \to \Pi, \mathbf{ProjHash}_{hp} : L \to \Pi_L$.

Informally, SPHF is defined by the following algorithms:

- **HashKG**(L): generates a hashing key hk for the language L;
- **ProjKG**(hk, L): derives the projection key hp from the hashing key hk; [1]
- **Hash**(hk, L, C): outputs the hash value of the world C from the hashing key hk;
- **ProjHash**(hp, L, C, w): outputs the hash value of the world C from the projection key hp, and the witness w that $C \in L$.

[1] In some special SPHF, the projection key may depend on the word $C \in L$.

Properties of SPHF

Normally, a SPHF has the following two properties:

- **Projection**: If a word $C \in L$ with w the witness, then

$$\textbf{Hash}(hk, L, C) = \textbf{ProjHash}(hp, L, C, w);$$

- **Smoothness**: If a word $C \in X/L$, then

$$(hp, \textbf{Hash}(hk, L, C)) \stackrel{s}{\equiv} (hp, R)$$

  where $\stackrel{s}{\equiv}$ means 'statistically indistinguishable', and $R \stackrel{\$}{\leftarrow} \Pi$ (hash value space).

Extension of the '**Smoothness**' Property:
<u>**Smoothness$_2$**</u>: If there is another word $C' \in X/L$, then

$$(hp, \textbf{Hash}(hk, L, C), \textbf{Hash}(hk, L, C')) \stackrel{s}{\equiv} (hp, R, R')$$

where $R, R' \stackrel{\$}{\leftarrow} \Pi$.

SPHF-1 for ECP Problem

SPHF-1 for ECP Problem

Domain X_{DDH}:

$$X_{DDH} = \{u_1, u_2 : \exists r_1, r_2 \ \ s.t. \ u_1 = g_1^{r_1}, u_2 = g_2^{r_2}\}$$

Language L_{DDH}:

$$L_{DDH} = \{u_1, u_2 : \exists r \ s.t. \ u_1 = g_1^{r}, u_2 = g_2^{r}\}$$

SPHF-1 for ECP Problem

Domain X_{DDH}:

$$X_{DDH} = \{u_1, u_2 : \exists r_1, r_2 \;\; s.t. \; u_1 = g_1^{r_1}, u_2 = g_2^{r_2}\}$$

Language L_{DDH}:

$$L_{DDH} = \{u_1, u_2 : \exists r \; s.t. \; u_1 = g_1^{r}, u_2 = g_2^{r}\}$$

Suppose the word $C = (X_1, X_2)$, then the SPHF on L_{DDH} is:

- **HashKG**(L_{DDH}): $hk = (b_1, b_2) \xleftarrow{\$} \mathbb{Z}_p^2$;
- **ProjKG**(hk, L_{DDH}): $hp = g_1^{b_1} g_2^{b_2}$;
- **Hash**(hk, L_{DDH}, C): $\pi_{hk} = X_1^{b_1} X_2^{b_2}$;
- **ProjHash**(hp, L_{DDH}, C, r): $\pi_{hp} = hp^r = (g_1^{b_1} g_2^{b_2})^r$.

SPHF-1 for ECP Problem

One can see that (from the analysis of ECP):

- **Projection** (for correctness): If $C = (X_1, X_2) \in L_{DDH}$ with r the witness, i.e., $C = (g_1^r, g_2^r)$ then

$$\pi_{hk} = X_1^{b_1} X_2^{b_2} = g_1^{b_1 r} g_2^{b_2 r} = \pi_{hp};$$

- **Smoothness** (for soundness): If $C \in X_{DDH}/L_{DDH}$, then

$$(hp, \pi_{hk}) \stackrel{s}{\equiv} (hp, R).$$

Smoothness$_2$ of SPHF-1?

Smoothness$_2$ of SPHF-1?

Now, suppose there is another word $C' \in X_{DDH}/L_{DDH}$, and

$$\pi'_{hk} \leftarrow \mathbf{Hash}(hk, L_{DDH}, C').$$

Question:

$$(hp, \pi_{hk}, \pi'_{hk}) \stackrel{s}{\equiv} (hp, R, R') \text{ ?}$$

Smoothness$_2$ of SPHF-1?

Now, suppose there is another word $C' \in X_{DDH}/L_{DDH}$, and

$$\pi'_{hk} \leftarrow \textbf{Hash}(hk, L_{DDH}, C').$$

Question:

$$(hp, \pi_{hk}, \pi'_{hk}) \overset{s}{\equiv} (hp, R, R') \text{ ?}$$

Answer:

$$(hp, \pi_{hk}, \pi'_{hk}) \overset{s}{\equiv} (hp, R, R') \ \times$$

No smoothness$_2$!

SPHF-2 for ECP Problem

Suppose that $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$, a collision-resistant hash function.

- **HashKG**(L_{DDH}): $hk = (a_1, a_2, b_1, b_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^4$;
- **ProjKG**(hk, L_{DDH}): $hp = (hp_1, hp_2) = (g_1^{a_1} g_2^{a_2}, g_1^{b_1} g_2^{b_2})$;
- **Hash**(hk, L_{DDH}, C): $\pi_{hk} = X_1^{a_1+\alpha b_1} X_2^{a_2+\alpha b_2}$, where $\alpha = H(X_1, X_2)$;
- **ProjHash**(hp, L_{DDH}, C, r): $\pi_{hp} = hp_1^r \cdot hp_2^{\alpha r}$.

SPHF-2 for ECP Problem

Suppose that $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$, a collision-resistant hash function.

- **HashKG**(L_{DDH}): $hk = (a_1, a_2, b_1, b_2) \xleftarrow{\$} \mathbb{Z}_p^4$;
- **ProjKG**(hk, L_{DDH}): $hp = (hp_1, hp_2) = (g_1^{a_1} g_2^{a_2}, g_1^{b_1} g_2^{b_2})$;
- **Hash**(hk, L_{DDH}, C): $\pi_{hk} = X_1^{a_1+\alpha b_1} X_2^{a_2+\alpha b_2}$, where $\alpha = H(X_1, X_2)$;
- **ProjHash**(hp, L_{DDH}, C, r): $\pi_{hp} = hp_1^r \cdot hp_2^{\alpha r}$.

One can see that :

- **Projection**: If $C = (X_1, X_2) \in L_{DDH}$, then

$$\pi_{hk} = \pi_{hp};$$

- **Smoothness**$_2$: If $C \in X_{DDH}/L_{DDH}$, $C' \in X_{DDH}/L_{DDH}$, then

$$(hp, \pi_{hk}, \pi'_{hk}) \stackrel{s}{\equiv} (hp, R, R')$$

Proof for 'Smoothness$_2$' of SPHF-2:

(Denote log $= log_{g_1}$ and suppose that log$(g_2) = w$, $C = (X_1, X_2) = (g_1^{r_1}, g_2^{r_2}), C' = (X_1', X_2') = (g_1^{r_1'}, g_2^{r_2'}), r_1 \neq r_2, r_1' \neq r_2'$.)

Proof for 'Smoothness$_2$' of SPHF-2:

(Denote $\log = log_{g_1}$ and suppose that $\log(g_2) = w$, $C = (X_1, X_2) = (g_1^{r_1}, g_2^{r_2})$, $C' = (X_1', X_2') = (g_1^{r_1'}, g_2^{r_2'})$, $r_1 \neq r_2, r_1' \neq r_2'$.)

Note that $hp = (hp_1, hp_2) = (g_1^{a_1} g_2^{a_2}, g_1^{b_1} g_2^{b_2})$ which constraints (a_1, a_2, b_1, b_2) to satisfy

$$a_1 + wa_2 = \log(hp_1) \tag{3}$$

$$b_1 + wb_2 = \log(hp_2) \tag{4}$$

Proof for 'Smoothness$_2$' of SPHF-2:

(Denote $\log = log_{g_1}$ and suppose that $\log(g_2) = w$, $C = (X_1, X_2) = (g_1^{r_1}, g_2^{r_2})$, $C' = (X_1', X_2') = (g_1^{r_1'}, g_2^{r_2'})$, $r_1 \neq r_2$, $r_1' \neq r_2'$.)

Note that $hp = (hp_1, hp_2) = (g_1^{a_1} g_2^{a_2}, g_1^{b_1} g_2^{b_2})$ which constraints (a_1, a_2, b_1, b_2) to satisfy

$$a_1 + wa_2 = \log(hp_1) \tag{3}$$

$$b_1 + wb_2 = \log(hp_2) \tag{4}$$

Moreover, π_{hk}, π'_{hk} constraint (a_1, a_2, b_1, b_2) to satisfy

$$r_1 a_1 + r_2 wa_2 + \alpha r_1 b_1 + \alpha r_2 wb_2 = \log(\pi_{hk}) \tag{5}$$

$$r_1' a_1 + r_2' wa_2 + \alpha' r_1' b_1 + \alpha' r_2' wb_2 = \log(\pi'_{hk}) \tag{6}$$

Equations (3),(4),(5),(6) are linearly independent regarding a_1, a_2, b_1, b_2. Hence, the distribution of (π_{hk}, π'_{hk}) is uniform in $\mathbb{G}$.

Membership Indistinguishable Language

Let X be a set and a language $L \subseteq X$. Suppose that word $C \xleftarrow{\$} L$ and word $C \xleftarrow{\$} X/L$. Then we say L is a membership indistinguishable language if,

$$(C) \stackrel{c}{\equiv} (C')$$

where $\stackrel{c}{\equiv}$ means 'computationally indistinguishable'.

Membership Indistinguishable Language

Let X be a set and a language $L \subseteq X$. Suppose that word $C \xleftarrow{\$} L$ and word $C \xleftarrow{\$} X/L$. Then we say L is a membership indistinguishable language if,

$$(C) \stackrel{c}{\equiv} (C')$$

where $\stackrel{c}{\equiv}$ means 'computationally indistinguishable'.

Language L_{DDH}:

$$L_{DDH} = \{u_1, u_2 : \exists r \ s.t. \ u_1 = g_1^r, u_2 = g_2^r\}$$

It is easy to see that the language L_{DDH} is a membership indistinguishable language following the DDH assumption.

Part III: Applications of SPHF

CCA-Secure PKE from SPHFs

Suppose that,

- HF_1=(**HashKG$_1$**,**ProjKG$_1$**,**Hash$_1$**,**ProjHash$_1$**):a smooth projective hash function;
- HF_2=(**HashKG$_2$**,**ProjKG$_2$**,**Hash$_2$**, **ProjHash$_2$**): a smooth projective hash function; (smoothness$_2$)
- Both SPHFs are for the same language L which is a membership indistinguishable language.

Generic Construction

KeyGen(λ): $\mathbf{HashKG}_1(L) \rightarrow hk_1, \mathbf{ProjKG}_1(hk_1, L) \rightarrow hp_1, \mathbf{HashKG}_2(L) \rightarrow hk_2, \mathbf{ProjKG}_2(hk_2, L) \rightarrow hp_2$ and set

$$pk = (hp_1, hp_2), sk = (hk_1, hk_2)$$

Enc(m): Pick $y \xleftarrow{\$} L$ together with a witness w. Compute

$$\pi_{hp_1} = \mathbf{ProjHash}_1(hp_1, L, y, w)$$

$$c = \pi_{hp_1} \oplus m$$

$$\pi_{hp_2} = \mathbf{ProjHash}_2(hp_2, L, (y, c), w)$$

Set the ciphertext as (y, c, π_{hp_2}).
Dec(y, c, π_{hp_2}): If $\mathbf{Hash}_2(hk_2, L, (y, c)) \neq \pi_{hp_2}$, output $\perp$. Otherwise, output

$$m = c \oplus \mathbf{Hash}_1(hk_1, L, y)$$

Construction of CCA-secure PKE from SPHF(Cont'd)

Security Analysis

1. Correctness:

$$c \oplus \textbf{Hash}_1(hk_1, L, y) = c \oplus \textbf{ProjHash}_1(hp_1, L, y, w) = c \oplus \pi_{hp_1} = m$$

2. Security against CCA

Hard Problem: Given the language L and y^*, decide whether $y^* \in L$ or not.

Reduction Map:

Simulator | **Adversary**

$pk = (hp_1, hp_2), sk = (hk_1, hk_2)$

$\xrightarrow{pk}$

Choose (y, c, π)

$\xleftarrow{(y,c,\pi)}$

If $\mathbf{Hash}_2(hk_2, L, y, c) = \pi$,
$m = c \oplus \mathbf{Hash}_1(hk_1, L, y)$;
Otherwise, return $\perp$.

Decryption Query I

$\xrightarrow{m/\perp}$

Choose m_0, m_1

$\xleftarrow{m_0, m_1}$

$b \xleftarrow{\$} \{0,1\}$,
$c^* = \mathbf{Hash}_1(hk_1, L, y^*) \oplus m_b$;
$\pi^* = \mathbf{Hash}_2(hk_2, L, (y^*, c^*))$.

$\xrightarrow{(y^*, c^*, \pi^*)}$

Choose (y, c, π),

$\xleftarrow{(y,c,\pi)}$

If $\mathbf{Hash}_2(hk_2, L, y, c) = \pi$,
$m = c \oplus \mathbf{Hash}_1(hk_1, L, y)$;
Otherwise, return $\perp$.

Decryption Query II

$\xrightarrow{m/\perp}$

$\xleftarrow{b'}$

If $b = b'$,output 1;
Otherwise, output 0.

Proof Analysis:

Case 1: $y^* \in L$. The simulation is indistinguishable from the actual attack;

Case 2: $y^* \in X/L$. For any decryption query input (y, c, π) where $y \notin L$, we should consider the following two cases:

- $(y, c) = (y^*, c^*)$ but $\pi \neq \pi^*$: Being rejected.
- $(y, c) \neq (y^*, c^*)$: By the smoothness$_2$ property of HF_2, the value π is random and independent to the adversary. That is, the adversary can only output the correct π with negligible probability.

Due to the smoothness property of HF_1, the value $\textbf{Hash}_1(hk_1, L, y^*)$ for $y^* \in X/L$ is uniformly random and hence perfectly hides the encrypted messages. (one-time pad)

Cramer-Shoup Encryption

Let $\sharp\mathbb{G} = p, g_1, g_2 \in \mathbb{G}$ and $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$.

- **KeyGen**: $sk = (\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2) \in \mathbb{Z}_p^6, pk = (g, h, u, v)$ where $h = g_1^{\alpha_1} g_2^{\alpha_2}, u = g_1^{\beta_1} g_2^{\beta_2}, v = g_1^{\gamma_1} g_2^{\gamma_2}$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT =< C_1, C_2, C_3, C_4 >=< g_1^r, g_2^r, h^r m, u^r v^{r\theta} >,$$

where $\theta = H(C_1, C_2, C_3)$.

- **Dec**$_{sk}(C_1, C_2, C_3, C_4)$: If $C_4 = C_1^{\beta_1+\theta\gamma_1} C_2^{\beta_2+\theta\gamma_2}$, where $\theta = H(C_1, C_2, C_3)$, output

$$m = C_3 / (C_1^{\alpha_1} \cdot C_2^{\alpha_2}),$$

otherwise output $\perp$.

Cramer-Shoup Encryption

Let $\sharp\mathbb{G} = p, g_1, g_2 \in \mathbb{G}$ and $H : \{0,1\}^* \to \mathbb{Z}_p$.

- **KeyGen**: $sk = (\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2) \in \mathbb{Z}_p^6$, $pk = (g, h, u, v)$ where $h = g_1^{\alpha_1} g_2^{\alpha_2}$, $u = g_1^{\beta_1} g_2^{\beta_2}$, $v = g_1^{\gamma_1} g_2^{\gamma_2}$.
- **Enc**$_{pk}(m)$: $r \leftarrow_R \mathbb{Z}_p$, output

$$CT =< C_1, C_2, C_3, C_4 >=< g_1^r, g_2^r, h^r m, u^r v^{r\theta} >,$$

  where $\theta = H(C_1, C_2, C_3)$.
- **Dec**$_{sk}(C_1, C_2, C_3, C_4)$: If $C_4 = C_1^{\beta_1+\theta\gamma_1} C_2^{\beta_2+\theta\gamma_2}$, where $\theta = H(C_1, C_2, C_3)$, output

$$m = C_3/(C_1^{\alpha_1} \cdot C_2^{\alpha_2}),$$

  otherwise output $\perp$.

Follow the framework using SPHF-1 and SPHF-2 on L_{DDH}!

2-Message Oblivious Transfer Protocol from SPHF

SPHF for Oblivious Transfer Construction

2-Message Oblivious Transfer Protocol from SPHF

Suppose that the sender takes as input a pair of strings γ_0, γ_1 and the receiver takes as input a choice bit b.

Sender (γ_0, γ_1)		**Receiver**
		$x_b \xleftarrow{\$} L, w_b$ is the witness
		$x_{1-b} \xleftarrow{\$} X/L$
	$\xleftarrow{\quad (x_0, x_1, L) \quad}$	
$\mathbf{HashKG}(L) = hk,$		
$\mathbf{ProjKG}(hk, L) = hp\,;$		
$y_0 = \gamma_0 \oplus \mathbf{Hash}(hk, L, x_0)\,;$		
$y_1 = \gamma_1 \oplus \mathbf{Hash}(hk, L, x_1)\,;$		
	$\xrightarrow{\quad (y_0, y_1, hp) \quad}$	
		$\gamma_b = y_b \oplus \mathbf{ProjHash}(hp, L, x_b, w_b)$

Here, the SPHF is on the language $L \subseteq X$ which is a membership indistinguishable language.

Security Analysis

- **Receiver Security.** Membership indistinguishable language; (x_b is indistinguishable from x_{1-b})
- **Sender Security.** Smoothness property; (y_{1-b} gives no information about γ_{1-b})
- **Malicious Receivers.** Might choose $x_0, x_1 \in L$? By requiring special word pair from L_{DDH}. (verifiable smoothness)

A Framework for PAKE from SPHF

SPHF for Password-based Authenticated Key Exchange

A Framework for PAKE from SPHF

Suppose that the parties take as input a shared password w.

$\mathbf{P}_1$		$\mathbf{P}_2$
$c_1 = C_\rho(w, r_1)$	$\xrightarrow{c_1}$	
		$\mathbf{HashKG}(L) \to hk_1$; $\mathbf{ProjKG}(hk_1, L, c_1) \to hp_1$; $c_2 = C_\rho(w, r_2)$
	$\xleftarrow{(hp_1, c_2)}$	
$\mathbf{HashKG}(L) \to hk_2$; $\mathbf{ProjKG}(hk_2, L, c_2) \to hp_2$;		
	$\xrightarrow{hp_2}$	
$SK_1 = \mathbf{ProjHash}(hp_1, L, (c_1, w), r_1) \oplus \mathbf{Hash}(hk_2, L, (c_2, w))$.		$SK_2 = \mathbf{Hash}(hk_1, L, (c_1, w)) \oplus \mathbf{ProjHash}(hp_2, L, (c_2, w), r_2)$.

Here, $C_\rho(w, r)$ is a commitment to w using random-coins r (witness) and common string ρ.

Security Analysis

- **Membership Indistinguishable Language.** Implied by the hiding property of commitment;
- **Passive Adversary.** Given (c_1, hp_1, c_2, hp_2),

$$(\textbf{Hash}(hk_2, L, (c_2, w)), \textbf{Hash}(hk_1, L, (c_1, w)) \stackrel{c}{\equiv} (R_1, R_2),$$

  where $(R_1, R_2) \xleftarrow{\$} \Pi^2$.
- **Adaptive Adversary.** Generates a commitment c' to a guessing password w', then $(c', w') \in X/L$ and thus from the view of adversary (who only see hp and not hk)

$$(\textbf{Hash}(hk, L, (c', w')) \stackrel{s}{\equiv} R,$$

  where $R \xleftarrow{\$} \Pi$.

More about SPHF

- **Construction**
  - Quadratic Assumption;
  - N-Residuosity Assumption;
  - Derived from CPA-PKE, CCA-PKE;
  - ...
- **More applications**
  - Extractable commitment;
  - Leakage-resilient PKE;
  - Lossy encryption;
  - Lossy trapdoor hash functions (LTDF);
  - ...

Thank you

Thank you

Any questions?