Extractable Witness Encryption for KZG Commitments and Efficient Laconic OT

Nils Fleischhacker, Mathias Hall-Andersen, and Mark Simkin

13. December 2024

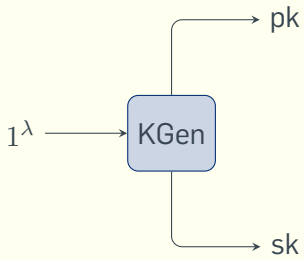
RUB

HG HORST GÖRTZ INSTITUT

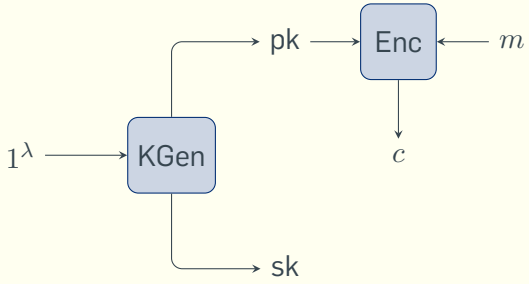
CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

Witness Encryption

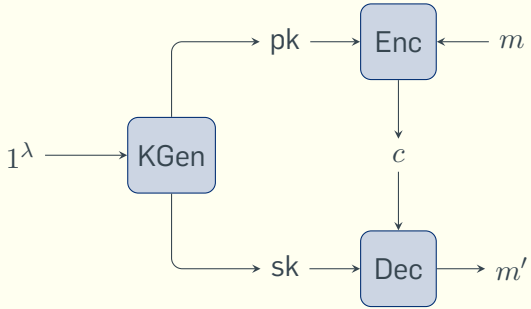
Witness Encryption



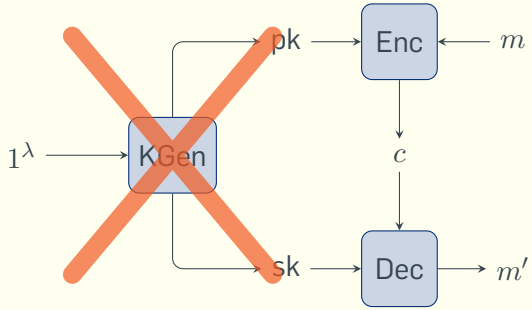
Witness Encryption



Witness Encryption



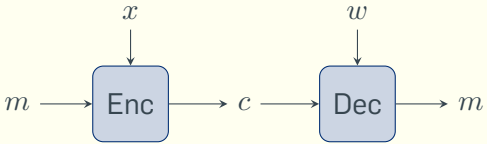
Witness Encryption



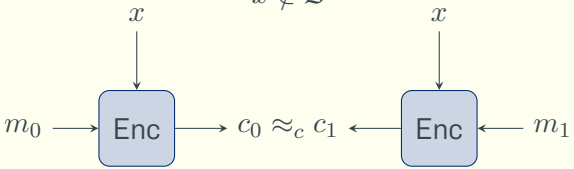
$$\mathcal{R} = \{(x, w) \mid M_{\mathcal{R}}(x, w) = 1\} \quad \mathcal{L} = \{x \mid \exists w.\ (x, w) \in \mathcal{R}\}$$

Witness Encryption

$(x, w) \in \mathcal{R}$



$x \notin \mathcal{L}$



KZG-Commitments

Setup$(1^\lambda, 1^d)$

$\tau \leftarrow \mathbb{F}_p$

return $([\tau^0]_1, \ldots, [\tau^d]_1, [1]_2, [\tau]_2)$

Commit(ck, f)

$$\mathtt{com} := \sum_{i=0}^{d} f_i \cdot [\tau^i]_1$$

return com

Open$(\mathsf{ck}, \mathtt{com}, f, \alpha, \beta)$

$$q(X) := \frac{f(X) - \beta}{X - \alpha}$$

$$\pi := \sum_{i=0}^{d} q_i \cdot [\tau^i]_1$$

return π

Verify$(\mathsf{ck}, \mathtt{com}, \pi, \alpha, \beta)$

return $e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$

KZG-Commitments

Setup$(1^\lambda, 1^d)$

$\tau \leftarrow \mathbb{F}_p$

return $([\tau^0]_1, \ldots, [\tau^d]_1, [1]_2, [\tau]_2)$

Commit(ck, f)

$\mathtt{com} := \sum_{i=0}^{d} f_i \cdot [\tau^i]_1$

return $\mathtt{com}$

Open$(\mathsf{ck}, \mathtt{com}, f, \alpha, \beta)$

$q(X) := \dfrac{f(X) - \beta}{X - \alpha}$

$\pi := \sum_{i=0}^{d} q_i \cdot [\tau^i]_1$

return π

Verify$(\mathsf{ck}, \mathtt{com}, \pi, \alpha, \beta)$

return $e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$

$$\mathcal{L}_{\mathsf{ck}} = \{(\mathtt{com}, \alpha, \beta) \mid \exists \pi.\ e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)\}$$

KZG-Commitments

Setup$(1^\lambda, 1^d)$

$\tau \leftarrow \mathbb{F}_p$

return $([\tau^0]_1, \ldots, [\tau^d]_1, [1]_2, [\tau]_2)$

Commit(ck, f)

$\mathtt{com} := \sum_{i=0}^{d} f_i \cdot [\tau^i]_1$

return $\mathtt{com}$

Open$(\mathsf{ck}, \mathtt{com}, f, \alpha, \beta)$

$q(X) := \dfrac{f(X) - \beta}{X - \alpha}$

$\pi := \sum_{i=0}^{d} q_i \cdot [\tau^i]_1$

return π

Verify$(\mathsf{ck}, \mathtt{com}, \pi, \alpha, \beta)$

return $e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$

$$\mathcal{L}_{\mathsf{ck}} = \{(\mathtt{com}, \alpha, \beta) \mid \exists \pi.\ e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)\} = \mathbb{G}_1 \times \mathbb{F}_q^2$$

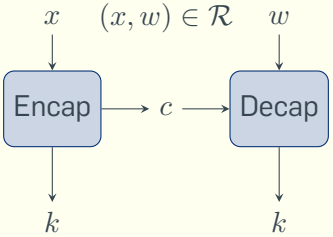
$$\pi := \left[\frac{f(\tau) - \beta}{\tau - \alpha}\right]_1$$

Extractable Witness Encryption

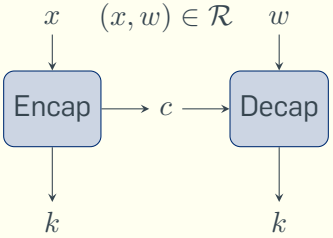
Definition (Extractability)

A witness encryption scheme is extractable, if there exists a PPT extractor Ext, such that for any PPT $\mathcal{A}$ that distinguishes $\mathsf{Enc}(x, m_0)$ from $\mathsf{Enc}(x, m_1)$ with non-negligible advantage, then Ext extracts a witness w for x with non-negligible probability.

Extractable Witness KEM



Extractable Witness KEM



Definition (Extractability)

A witness KEM is extractable, if there exists a PPT extractor Ext, such that for any PPT $\mathcal{A}$ that for $(c, k_0) \leftarrow \text{Encap}(x)$ and $k_1 \leftarrow \{0,1\}^\lambda$ distinguishes (c, k_0) from (c, k_1) with non-negligible advantage, then Ext extracts a witness w for x with non-negligible probability.

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \overset{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

$\mathsf{Encap}^{\mathsf{H}}(\mathsf{ck}, (\mathtt{com}, \alpha, \beta))$	$\mathsf{Decap}^{\mathsf{H}}(\mathsf{ck}, \pi, \mathtt{ct})$
	return k
return $(\mathtt{ct}, \mathsf{k})$	

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \overset{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

$\mathsf{Encap}^{\mathsf{H}}(\mathsf{ck}, (\mathtt{com}, \alpha, \beta))$	$\mathsf{Decap}^{\mathsf{H}}(\mathsf{ck}, \pi, \mathtt{ct})$
$r \leftarrow \mathbb{F}_p$	
$s := e(r \cdot (\mathtt{com} - [\beta]_1), [1]_2)$	
	return k
return $(\mathtt{ct}, \mathsf{k})$	

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \overset{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

$\mathsf{Encap}^{\mathsf{H}}(\mathsf{ck}, (\mathtt{com}, \alpha, \beta))$	$\mathsf{Decap}^{\mathsf{H}}(\mathsf{ck}, \pi, \mathtt{ct})$
$r \leftarrow \mathbb{F}_p$	
$s := e(r \cdot (\mathtt{com} - [\beta]_1), [1]_2)$	
	return k
$\mathsf{k} := \mathsf{H}(s)$	
return $(\mathtt{ct}, \mathsf{k})$	

An Extractable Witness KEM for KZG

$$e(\mathtt{com} - [\beta]_1, [1]_2) \overset{?}{=} e(\pi, [\tau]_2 - [\alpha]_2)$$

$\mathsf{Encap}^{\mathsf{H}}(\mathsf{ck}, (\mathtt{com}, \alpha, \beta))$	$\mathsf{Decap}^{\mathsf{H}}(\mathsf{ck}, \pi, \mathtt{ct})$
$r \leftarrow \mathbb{F}_p$	$s := e(\pi, \mathtt{ct})$
$s := e(r \cdot (\mathtt{com} - [\beta]_1), [1]_2)$	$\mathsf{k} := \mathsf{H}(s)$
$\mathtt{ct} := r \cdot ([\tau]_2 - [\alpha]_2)$	**return** k
$\mathsf{k} := \mathsf{H}(s)$	
return $(\mathtt{ct}, \mathsf{k})$	

Proving Extractability

The Assumption

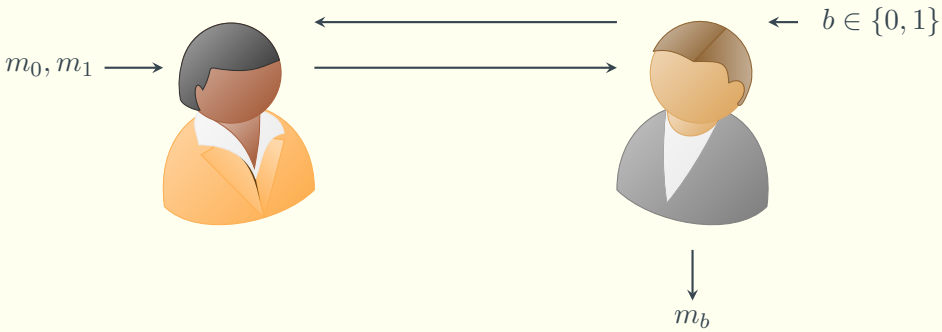
Definition (ℓ-DLOG)

The ℓ-DLOG problem is hard, if for any PPT $\mathcal{A}$ it holds that

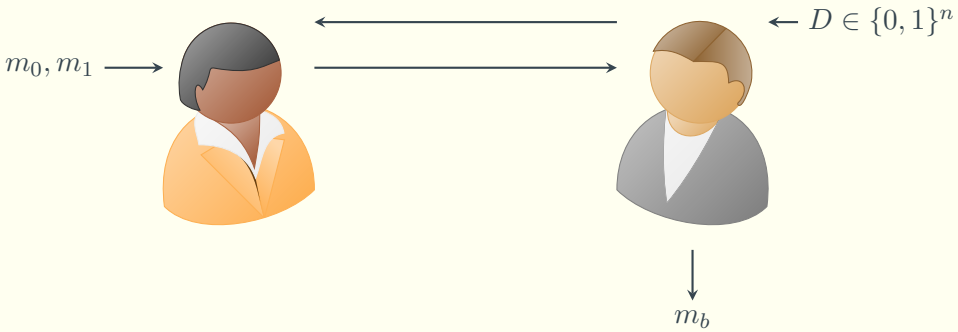
$$\Pr\left[\tau = \tau' : \begin{array}{r} \tau \leftarrow \mathbb{F}_p \\ \tau' \leftarrow \mathcal{A}(\mathsf{par}, ([\tau^0]_1, \ldots, [\tau^\ell]_1, [1]_2, [\tau]_2)) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

We prove extractability in the combined AGM and ROM.

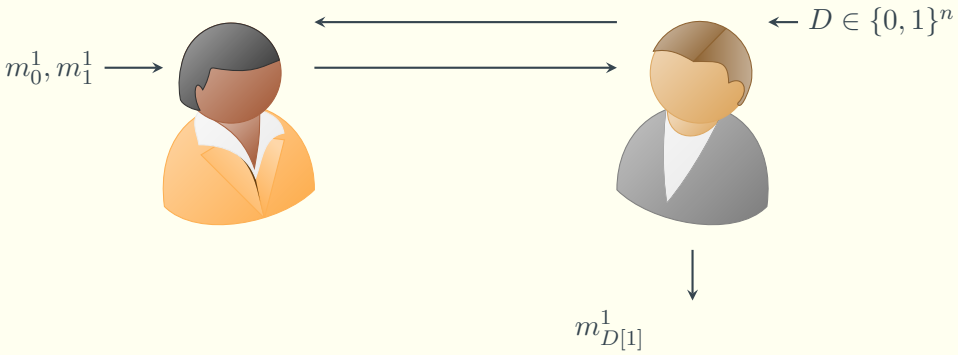
Laconic Oblivious Transfer [CDG+17]



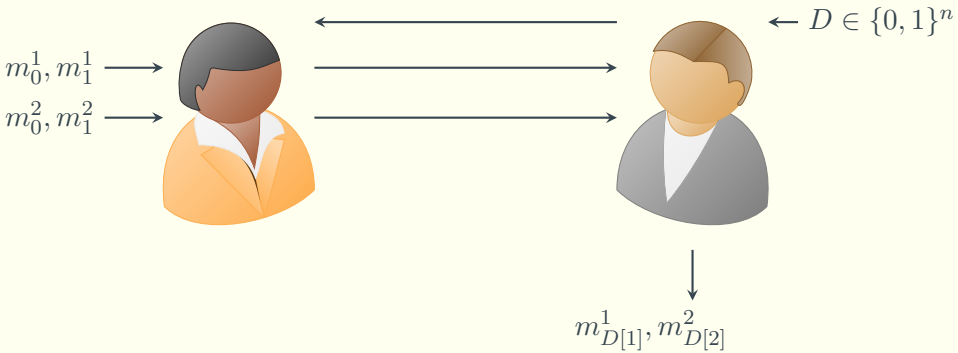
Laconic Oblivious Transfer [CDG+17]



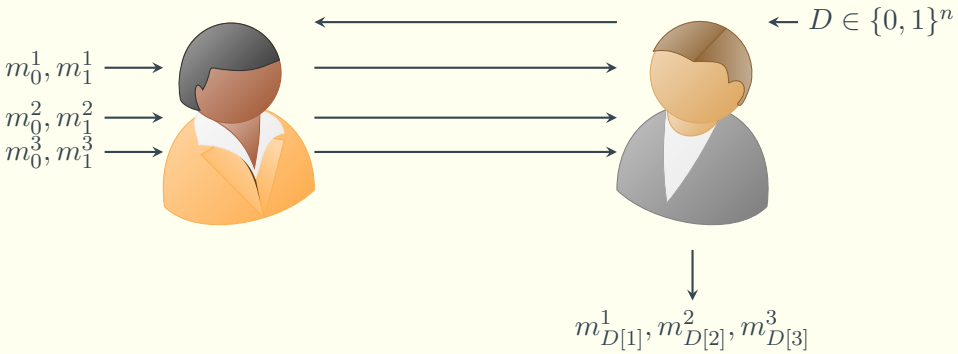
Laconic Oblivious Transfer [CDG+17]



Laconic Oblivious Transfer [CDG+17]



Laconic Oblivious Transfer [CDG+17]



Laconic OT from Witness Encryption

$\underline{\mathsf{Setup}(1^\lambda, 1^n)}$

$\mathtt{pp} \leftarrow \mathsf{VC.Setup}(1^\lambda, 1^n)$

$\underline{\mathsf{H}(\mathtt{pp}, D)}$

$(\mathtt{com}, \overline{\mathtt{aux}}) \leftarrow \mathsf{VC.Commit}(\mathtt{pp}, D)$

$(\pi_1, \ldots, \pi_n) \leftarrow \mathsf{BatchOpen}(\mathtt{pp}, \overline{\mathtt{aux}})$

return $(\mathtt{com}, (D, \pi_1, \ldots, \pi_n))$

$\underline{\mathsf{Send}(\mathtt{pp}, \mathtt{digest}, i, m_0, m_1)}$

$\mathtt{ct}_0 \leftarrow \mathsf{WE.Enc}(\mathtt{pp}, (\mathtt{digest}, i, 0), m_0)$

$\mathtt{ct}_1 \leftarrow \mathsf{WE.Enc}(\mathtt{pp}, (\mathtt{digest}, i, 1), m_1)$

return $(\mathtt{ct}_0, \mathtt{ct}_1)$

$\underline{\mathsf{Receive}(\mathtt{pp}, \mathtt{aux}, (\mathtt{ct}_0, \mathtt{ct}_1), i)}$

$b := D_i$

$m_b \leftarrow \mathsf{WE.Dec}(\mathtt{pp}, \pi_i, \mathtt{ct}_b)$

return m_b

Benchmarks

$\lvert D\rvert$	Sizes			Times		
	`pp`	`digest`	Sender Msg.	Hash	Send	Receive
2^{6}	3.2 KB	48 B	256 B	173 ms	4 ms	1 ms
2^{8}	12.2 KB	48 B	256 B	723 ms	4 ms	1 ms
2^{10}	48.2 KB	48 B	256 B	3 s	4 ms	1 ms
2^{12}	192.2 KB	48 B	256 B	10 s	4 ms	1 ms
2^{14}	768.2 KB	48 B	256 B	43 s	4 ms	1 ms
2^{16}	3.0 MB	48 B	256 B	3 min	5 ms	1 ms
2^{18}	12.0 MB	48 B	256 B	8 min	5 ms	1 ms
2^{31}	96.0 GB	48 B	256 B	—	5 ms	1 ms