

# MINICRYPT WITH ALGEBRAIC STRUCTURE

ADVANCED TOPICS IN CYBERSECURITY CRYPTOGRAPHY (7CCSMATC)

---

Martin R. Albrecht

# OUTLINE

Introduction

Computational Assumptions

(Bounded) Homomorphic One-way Functions (HOWFs)

(Bounded) Input-Homomorphic Weak Pseudorandom Functions (IHwPRFs)

## INTRODUCTION

---

## MAIN REFERENCE

Navid Alammati, Hart Montgomery, Sikhar Patranabis, and Arnab Roy. **Minicrypt Primitives with Algebraic Structure and Applications.** In: *Journal of Cryptology* 36.1 (Jan. 2023), p. 2.

DOI: [10.1007/s00145-022-09442-2](https://doi.org/10.1007/s00145-022-09442-2)



## MINICRYPT: ONE-WAY FUNCTIONS ALL THE WAY DOWN

- OWFs  $\Rightarrow$  PRGs [BM84]; PRGs  $\Rightarrow$  PRFs [GGM86]; PRFs  $\Rightarrow$  PRPs [LR88]
- OWFs  $\Rightarrow$  Digital Signatures [Lam79; Mer79; NY89]

[AMPR23]

“It is natural to ask: is there any sort of mathematical structure that is inherent to [Cryptomania] primitives as well?”

## MINICRYPT: ONE-WAY FUNCTIONS ALL THE WAY DOWN

- OWFs  $\Rightarrow$  PRGs [BM84]; PRGs  $\Rightarrow$  PRFs [GGM86]; PRFs  $\Rightarrow$  PRPs [LR88]
- OWFs  $\Rightarrow$  Digital Signatures [Lam79; Mer79; NY89]

### Question

Is it possible to construct Cryptomania primitives from simple Minicrypt primitives that are additionally equipped with some **algebraic structure**?

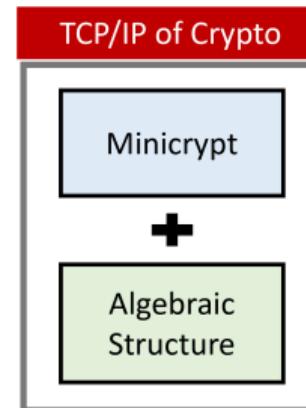
# ALTERNATIVE PITCH I



## ALTERNATIVE PITCH II

### A “Narrow-Waist” Approach

Assumptions
<ul style="list-style-type: none"><li>• Discrete log</li><li>• Search LWE</li><li>• Square-root finding</li></ul>
<ul style="list-style-type: none"><li>• CDH</li><li>• RSA</li></ul>
<ul style="list-style-type: none"><li>• DDH/MDDH/D-Lin</li><li>• Decision LWE</li><li>• QR/DCR</li><li>• Approx. GCD</li><li>• Hidden number problem</li><li>• Finite Field isomorphism</li></ul>



Primitives
<ul style="list-style-type: none"><li>• Schnorr-style signatures</li><li>• Trapdoor CRHFs</li><li>• Succinct commitments</li></ul>
<ul style="list-style-type: none"><li>• Two-Party NIKE</li><li>• CPA-Secure PKE</li><li>• Trapdoor Functions</li><li>• IBE</li><li>• KDM-Secure PKE</li><li>• DV-NIZK</li></ul>
<ul style="list-style-type: none"><li>• PIR</li><li>• Lossy TDFs</li><li>• OT and MPC</li></ul>

## COMPUTATIONAL ASSUMPTIONS

---

## DIFFIE-HELLMAN (DH)

### Discrete Logarithms

Let  $p$  be a  $\lambda$ -bit prime and let  $g$  be a generator of the multiplicative group  $\mathbb{Z}_p^*$ . Given  $g^a \bmod p$  find  $a$ .

### DH

Let  $p$  be a  $\lambda$ -bit prime and let  $g$  be a generator of the multiplicative group  $\mathbb{Z}_p^*$ . Given

$$(g^a \bmod p, g^b \bmod p, u),$$

decide if  $u \equiv g^{ab} \bmod p$  or random.

# LEARNING WITH ERRORS (LWE)

## Definition (LWE)

Let  $m, n, q$  be positive integers,  $\chi$  be a probability distribution on  $\mathbb{Z}$  and  $\mathbf{s}$  be a uniformly random vector in  $\mathbb{Z}_q^n$ . Denote by  $\mathcal{L}_{\mathbf{s}, \chi}$  the probability distribution on  $\mathbb{Z}_q^n \times \mathbb{Z}_q^m$  obtained by choosing  $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$  uniformly at random, choosing every component of  $\mathbf{e} \in \mathbb{Z}$  according to  $\chi$  and returning

$$(\mathbf{A}, \mathbf{c}) = (\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m.$$

**Search-LWE** recover  $\mathbf{s}$  from  $(\mathbf{A}, \mathbf{c}) = (\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$  sampled according to  $\mathcal{L}_{\mathbf{s}, \chi}$ .

**Decision-LWE** decide if  $(\mathbf{A}, \mathbf{c})$  is sampled from  $\mathcal{L}_{\mathbf{s}, \chi}$  or the uniform distribution.

Search-LWE is like DLOG, Decision-LWE is like DH

## DH AND LWE

DH Land	LWE Land
$g$	$\mathbf{A}$
$g^x$	$\mathbf{A} \cdot \mathbf{s} + \mathbf{e}$
$g^x \cdot g^y = g^{x+y}$	$(\mathbf{A} \cdot \mathbf{s} + \mathbf{e}_0) + (\mathbf{A} \cdot \mathbf{t} + \mathbf{e}_1) = \mathbf{A} \cdot (\mathbf{s} + \mathbf{t}) + \mathbf{e}'$
$(g^a)^b = (g^b)^a$	$\mathbf{t}^T \cdot (\mathbf{A} \cdot \mathbf{s} + \mathbf{e}) = \mathbf{t}^T \cdot \mathbf{A} \cdot \mathbf{s} + \mathbf{t}^T \cdot \mathbf{e}$ $\approx \mathbf{t}^T \cdot \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \cdot \mathbf{s}$
$\approx_c (g, g^a, g^b, g^{ab})$	$(\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e}, \mathbf{t}^T \cdot \mathbf{A} + \mathbf{d}^T, \mathbf{t}^T \cdot \mathbf{A} \cdot \mathbf{s} + \mathbf{e}')$
$\approx_c (g, g^a, g^b, u)$	$\approx_c (\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e}, \mathbf{t}^T \cdot \mathbf{A} + \mathbf{d}^T, u)$

## (BOUNDED) HOMOMORPHIC ONE-WAY FUNCTIONS (HOWFs)

---

# ONE-WAY FUNCTIONS (OWFs)

## Definition (OWF)

A function  $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$  is **one-way** if the following two conditions hold:

1. **(Easy to compute)** There exists a polynomial-time algorithm  $\mathcal{A}_f(x)$  computing  $f$ , i.e.  $\mathcal{A}_f(x) = f(x)$  for all  $x$ .
2. **(Hard to invert)** for every probabilistic polynomial-time  $\mathcal{B}$ , we have:

$$\Pr_{x \leftarrow \{0,1\}^n} [\mathcal{B}(1^n, f(x)) \in f^{-1}(f(x))] < 1/\text{poly}(n).$$

# (BOUNDED) HOMOMORPHIC ONE-WAY FUNCTIONS (HOWFs)

## Definition (Homomorphic OWF)

A homomorphic one-way function (HOWF) over an input group  $(\mathcal{X}, \oplus)$  and an output group  $(\mathcal{Y}, \otimes)$  is a one-way function  $f : \mathcal{X} \rightarrow \mathcal{Y}$  such that for any  $x_0, x_1 \in \mathcal{X}$ , we have  $f(x_0 \oplus x_1) = f(x_0) \otimes f(x_1)$ .

## Definition ( $\gamma$ -Bounded Homomorphic OWF)

A  $\gamma$ -bounded homomorphic one-way function ( $\gamma$ -bounded HOWF) over an input group  $(\mathcal{X}, \oplus)$  and an output group  $(\mathcal{Y}, \otimes)$  is a one-way function  $f : \mathcal{X} \rightarrow \mathcal{Y}$  such that

$$\mathcal{R} \left( f \left( \bigoplus_{j \in \{0, \dots, L-1\}} x_j \right) \right) = \mathcal{R} \left( \bigotimes_{j \in \{0, \dots, L-1\}} f(x_j) \right)$$

subject to the restriction  $L \leq \gamma$  and some  $\mathcal{R}$ .<sup>1</sup>

---

<sup>1</sup>This definition differs from Def. 3.2 in [AMPR23]. Additionally, we should specify the groups as efficiently computable.

# INSTANTIATION

From DLOG

$$\begin{aligned}f_g(x) &= g^x \\f_g(x_0 + x_1) &= g^{x_0} \cdot g^{x_1} \\&= f_g(x_0) \cdot f_g(x_1)\end{aligned}$$

From Search-LWE

$$\begin{aligned}\mathcal{R}(b \in \mathbb{Z}_q) &= \begin{cases} 0 & |b| \leq q/4 \\ 1 & |b| > q/4 \end{cases} \\f_A(s, e) &= A \cdot s + e \\ \mathcal{R}(f_A(s + t, e + d)) &= A \cdot s + e + A \cdot t + d \\&= \mathcal{R}(f_A(s, e) + f_A(t, d))\end{aligned}$$

Here be dragons!

# COLLISION RESISTANT HASH FUNCTIONS

## Definition (Collision Resistance)

A function  $f : \{0, 1\}^{\text{poly}(n)} \rightarrow \{0, 1\}^n$  is called collision resistant if for every probabilistic polynomial-time  $\mathcal{A}$ , we have

$$\Pr [x_0 \neq x_1 \wedge f(x_0) = f(x_1) \mid x_0, x_1 \leftarrow \mathcal{A}(f)] < 1/\text{poly}(n).$$

# HOWF $\Rightarrow$ CRHF

Setup:

1. Sample  $2n$  uniform elements in  $\mathcal{X}$ :

$$\mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow_{\$} \mathcal{X}^{2 \times n}$$

2. Compute

$$\mathbf{Y} := \begin{pmatrix} y_{0,0} & \dots & y_{0,n-1} \\ y_{1,0} & \dots & y_{1,n-1} \end{pmatrix} = \begin{pmatrix} f(x_{0,0}) & \dots & f(x_{0,n-1}) \\ f(x_{1,0}) & \dots & f(x_{1,n-1}) \end{pmatrix}$$

Call:

$$H_Y(\mathbf{r} = (r_0, \dots, r_{n-1})) = \bigotimes_{0 \leq j < n} y_{r_j, j}.$$

## PROOF I

We assume there exists some adversary  $\mathcal{A}$  that breaks collision resistance, i.e. outputs  $\mathbf{r}, \mathbf{r}'$  s.t.  $\mathbf{r} \neq \mathbf{r}'$  but  $H_Y(\mathbf{r}) = H_Y(\mathbf{r}')$ .

We construct an adversary  $\mathcal{B}$  that breaks the one-wayness of  $f(\cdot)$  using  $\mathcal{A}$  as a subroutine.

1.  $\mathcal{B}$  receives a challenge  $y^* \in \mathcal{Y}$  and wants to find  $x^*$  s.t.  $f(x^*) = y^*$ .
2.  $\mathcal{B}$  samples  $2n$  uniform elements in  $\mathcal{X}$ :  $\mathbf{X} := \{x_{b,j} \leftarrow_{\$} \mathcal{X}\}_{0 \leq j < n, b \in \{0,1\}}$  and sets  $\mathbf{Y} := \{y_{b,j} := f(x_{j,b})\}_{0 \leq j < n, b \in \{0,1\}}$
3.  $\mathcal{B}$  picks a random  $i \in \{0, \dots, n - 1\}$  and random  $b^* \in \{0, 1\}$  and sets  $y_{b^*,i} := y^*$ .

## PROOF II

4.  $\mathcal{B}$  runs  $\mathcal{A}$  on  $Y$  and receives  $r \neq r'$  back, s.t.  $H_Y(r) = H_Y(r')$ .
5. If  $r_i = r'_i$  then  $\mathcal{B}$  aborts (or outputs a random answer)

## PROOF III

6. Assume  $r_i = b^*$  (swap  $r_i$  and  $r'_i$  otherwise)

7. We have

$$\begin{aligned} \bigotimes_{0 \leq j < n} y_{r_j, j} &= \bigotimes_{0 \leq j < n} y_{r'_j, j} \\ \left( \bigotimes_{0 \leq j < i} y_{r_j, j} \right)^{-1} \cdot \bigotimes_{0 \leq j < n} y_{r_j, j} \cdot \left( \bigotimes_{i+1 \leq j < n} y_{r_j, j} \right)^{-1} &= \left( \bigotimes_{0 \leq j < i} y_{r_j, j} \right)^{-1} \cdot \bigotimes_{0 \leq j < n} y_{r'_j, j} \cdot \left( \bigotimes_{i+1 \leq j < n} y_{r_j, j} \right)^{-1} \\ y^* &= \left( \bigotimes_{0 \leq j < i} y_{r_j, j} \right)^{-1} \cdot \bigotimes_{0 \leq j < n} y_{r'_j, j} \cdot \left( \bigotimes_{i+1 \leq j < n} y_{r_j, j} \right)^{-1} \end{aligned}$$

8. Note that the RHS is independent of  $y^*$

## PROOF IV

9.  $\mathcal{B}$  outputs:

$$x^* = \left( \bigoplus_{0 \leq j < i} x_{r_j, j} \right)^{-1} \cdot \bigoplus_{0 \leq j < n} x_{r'_j, j} \cdot \left( \bigoplus_{i+1 \leq j < n} x_{r_j, j} \right)^{-1}$$

10. By the input-homomorphism, we have  $f(x^*) = y^*$ .
11.  $\mathcal{B}$  succeeds whenever  $\mathcal{A}$  succeeds and  $r_i \neq r'_i$ , which happens with probability at least  $1/n$  for randomly chosen  $i$ .

# SCHNORR-STYLE SIGNATURES

Let

- $f : \mathcal{X} \rightarrow \mathcal{Y}$  be a HOWF
- $n$  be some integer
- $H : \mathcal{Y} \times \{0,1\}^* \rightarrow \{0,1\}^n$  be a hash function

## Gen

- $\text{sk} := \mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow_{\$} \mathcal{X}^{2 \times n};$
- $\text{vk} := \mathbf{Y} := \begin{pmatrix} f(x_{0,0}) & \dots & f(x_{0,n-1}) \\ f(x_{1,0}) & \dots & f(x_{1,n-1}) \end{pmatrix}$

## Sign( $\text{sk}, m$ )

1.  $x^* \leftarrow_{\$} \mathcal{X}$  and set  $y^* \leftarrow f(x^*)$
2.  $\mathbf{r} := H(y^*, m)$
3.  $\hat{x} := x^* \oplus \left( \bigoplus_{0 \leq j < n} x_{r_j,j} \right)^{-1}$   
 $\sigma := (\mathbf{r}, \hat{x}, y^*)$

## Verify( $\text{vk}, \sigma, m$ )

Check:

1.  $\mathbf{r} \stackrel{?}{=} H(y^*, m)$
2.  $y^* \stackrel{?}{=} f(\hat{x}) \otimes \left( \bigotimes_{0 \leq j < n} y_{r_j,j} \right)$

# SCHNORR-STYLE SIGNATURES (DLOG INSTANTIATION: $f : x \rightarrow g^x \bmod p$ )

Let

- $n$  be some integer
- $H : \mathbb{Z}_p^* \times \{0,1\}^* \rightarrow \{0,1\}^n$  be a hash function

## Gen

- $\text{sk} := \mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow \mathbb{Z}_{p-1}^{2 \times n};$
- $\text{vk} := \mathbf{Y} := \begin{pmatrix} g^{x_{0,0}} & \dots & g^{x_{0,n-1}} \\ g^{x_{1,0}} & \dots & g^{x_{1,n-1}} \end{pmatrix}$

## Sign( $\text{sk}, m$ )

You will do this in Assignment 2.

## Verify( $\text{vk}, \sigma, m$ )

You will do this in Assignment 2.

# ACTUAL SCHNORR SIGNATURES

Let

- $H : \mathbb{G} \times \{0,1\}^* \rightarrow \mathbb{Z}_p$  be a hash function

## Gen

- $\text{sk} := x \leftarrow \mathbb{Z}_p$ ;
- $\text{vk} := Y := g^x$

Claus-Peter Schnorr. **Efficient Identification and Signatures for Smart Cards.** In: CRYPTO'89. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, New York, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0\_22

## Sign( $\text{sk}, m$ )

1.  $x^* \leftarrow \mathbb{Z}_p$  and set  $y^* \leftarrow g^{x^*}$
2.  $r := H(y^*, m)$
3.  $\hat{x} := x^* - x \cdot r$

$$\sigma := (r, \hat{x}, y^*)$$

## Verify( $\text{vk}, \sigma, m$ )

Check:

1.  $r \stackrel{?}{=} H(y^*, m)$
2.  $y^* \stackrel{?}{=} g^{\hat{x}} \cdot y^r = g^{\hat{x}} \cdot g^{xr}$

## PROOF

The proof of this (ubiquitous) construction relies on two proof techniques we have yet to cover:

- the Random Oracle Model
- Rewinding

*Note 3.7.* The aforementioned signature scheme has an a priori bounded number of homomorphic operations, which allows it to be instantiated similarly using a  $\gamma$ -bounded HOWF, subject to the restriction that  $2n < \gamma$ . — [AMPR23]

Here be dragons!

(BOUNDED) INPUT-HOMOMORPHIC  
WEAK PSEUDORANDOM FUNCTIONS  
(IHwPRFs)

---

# STATISTICAL DISTANCE (SD) AKA TOTAL VARIATION DISTANCE

## Definition (Statistical Distance)

Let  $\mathcal{D}_0$  and  $\mathcal{D}_1$  be two distributions with support  $\mathcal{X}$ . The Statistical Distance (SD) between  $\mathcal{D}_0$  and  $\mathcal{D}_1$  is:

$$\text{SD}(\mathcal{D}_0, \mathcal{D}_1) := 1/2 \cdot \sum_{x \in \mathcal{X}} |\mathcal{D}_0(x) - \mathcal{D}_1(x)|$$

## Lemma (Distinguishing)

*Given  $\text{poly}(\lambda)$  samples, no adversary can distinguish  $\mathcal{D}_0, \mathcal{D}_1$  with advantage  $\geq 1/\text{poly}(\lambda)$  when  $\text{SD}(\mathcal{D}_0, \mathcal{D}_1) < 1/\text{poly}(\lambda)$ .*

## LEFTOVER HASH LEMMA (LHL)

### Lemma (Leftover Hash Lemma)

Let  $(\mathcal{X}, \oplus)$  be a finite group of size  $|\mathcal{X}|$  and let  $n$  be a positive integer. Denote by  $\mathcal{U}(\mathcal{X})$  the uniform distribution on  $\mathcal{X}$ . For any fixed  $(2 \times n)$ -matrix of group elements

$\mathbf{X} := \{x_{b,j}\}_{b \in \{0,1\}, 0 \leq j < n} \in \mathcal{X}^{2 \times n}$  let:  $\mathcal{S}_{\mathbf{X}} = \left\{ \bigoplus_{0 \leq j < n} x_{j,r_j} : \mathbf{r} \leftarrow \{0,1\}^n \right\}$ . For sufficiently large  $n$  it holds that

$$\Pr_{\mathbf{X} \leftarrow \mathcal{X}^{2 \times n}} \left[ \text{SD}(\mathcal{S}_{\mathbf{X}}, \mathcal{U}(\mathcal{X})) > \sqrt[4]{\frac{|\mathcal{X}|}{2^n}} \right] \leq \sqrt[4]{\frac{|\mathcal{X}|}{2^n}}.$$

If  $n > \log(|\mathcal{X}|) + \omega(\log(\lambda))$  and  $\mathbf{X}$  uniform, then with overwhelming probability the statistical distance is negligible.

## Definition (PRF)

A PRF is a keyed function  $F_k : \mathcal{X} \rightarrow \mathcal{Y}$  with  $k \leftarrow_{\$} \mathcal{K}$ . We say  $F_k$  is  $(t, \varepsilon)$ -secure PRF if for Game<sub>0</sub> and Game<sub>1</sub> defined below we have:

$$\forall \mathcal{D} \in t \text{ steps: } \text{Adv}_F^{\text{prf}}(\mathcal{D}) = |\Pr[\mathcal{D}^{\text{Game}_1} = 1] - \Pr[\mathcal{D}^{\text{Game}_0} = 1]| < \varepsilon$$

Game <sub>0</sub>	F(x)
1: $f \leftarrow \emptyset$	1: <b>if</b> $x \in \pi.\text{keys}$ <b>then</b> $y \leftarrow f[x]$
2: <b>return</b> $\mathcal{D}^F$	2: <b>else</b>
Game <sub>1</sub>	
1: $f \leftarrow \emptyset; k \leftarrow_{\$} \mathcal{K}$	3: $y \leftarrow_{\$} \mathcal{Y}; f[x] \leftarrow y$
2: <b>return</b> $\mathcal{D}^F$	4: $y \leftarrow F_k(x) // \text{Game}_1$
	5: <b>return</b> $y$

# WEAK PRF

## Definition (wPRF)

A PRF is a keyed function  $F_k : \mathcal{X} \rightarrow \mathcal{Y}$  with  $k \leftarrow_{\$} \mathcal{K}$ . We say  $F_k$  is  $(t, \varepsilon)$ -secure wPRF if for Game<sub>0</sub> and Game<sub>1</sub> defined below we have:

$$\forall \mathcal{D} \in t \text{ steps: } \text{Adv}_F^{\text{wprf}}(\mathcal{D}) = |\Pr[\mathcal{D}^{\text{Game}_1} = 1] - \Pr[\mathcal{D}^{\text{Game}_0} = 1]| < \varepsilon$$

Game <sub>0</sub>	F()
1: $f \leftarrow \emptyset$	1: $x \leftarrow_{\$} \mathcal{X}$
2: <b>return</b> $\mathcal{D}^F$	2: <b>if</b> $x \in \pi.\text{keys}$ <b>then</b> $y \leftarrow f[x]$
Game <sub>1</sub>	3: <b>else</b>
1: $f \leftarrow \emptyset; k \leftarrow_{\$} \mathcal{K}$	4: $y \leftarrow_{\$} \mathcal{Y}; f[x] \leftarrow y$
2: <b>return</b> $\mathcal{D}^F$	5: $y \leftarrow F_k(x) // \text{Game}_1$
	6: <b>return</b> $x, y$

## (BOUNDED) IHwPRF

### Definition (Input-Homomorphic Weak PRF)

We call a keyed function  $F_k : \mathcal{X} \rightarrow \mathcal{Y}$  an IHwPRF if the following conditions hold:

1.  $F_k(\cdot)$  is a weak PRF.
2. Both  $(\mathcal{X}, \oplus)$  and  $(\mathcal{Y}, \otimes)$  are efficiently samplable groups.
3. The group operations  $\oplus$  and  $\otimes$ , and their inverses are efficiently computable.
4. For every  $k \in \mathcal{K}$  and every  $x_0, x_1 \in \mathcal{X}$ , we have:

$$F_k(x_0 \oplus x_1) = F_k(x_0) \otimes F_k(x_1).$$

### $\gamma$ -Bounded Variant

$$\mathcal{R} \left( F_k \left( \bigoplus_{0 \leq i < L} x_i \right) \right) = \mathcal{R} \left( \bigotimes_{0 \leq i < L} F_k(x_i) \right) \text{ for } L \leq \gamma \text{ and some } \mathcal{R}.$$

# IHWPRF FROM DH

$$F_k(h) := h^k$$

## BOUNDED IHwPRF FROM LWE

- Let  $D_\chi()$  be an algorithm taking  $\ell$  bits and outputting a sample following the distribution  $\chi$
- Let  $F' : \mathbb{Z}_q^n \rightarrow \{0, 1\}^\ell$  be a weak PRF outputting  $\ell$  bits
- Define the bounded IHwPRF as:

$$F_{(k,s)}(\mathbf{a}) = \langle \mathbf{a}, \mathbf{s} \rangle + D_\chi(F'_k(\mathbf{a})).$$

- Define the relation as

$$\mathcal{R}(b \in \mathbb{Z}_q) = \begin{cases} 0 & |b| \leq q/4 \\ 1 & |b| > q/4 \end{cases}$$

Here be dragons!

## OWFs FROM IHwUF/IHwPRF

Construction:

Let  $F_k : \mathcal{X} \rightarrow \mathcal{Y}$  be an IHwPRF. Fix  $n > 3 \log |\mathcal{X}|$  and sample

$$X := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow \$_{\mathcal{X}}^{2 \times n}$$

Define  $OWF_X : \{0,1\}^n \rightarrow \mathcal{X}$  as

$$OWF_X(r) = \bigoplus_{0 \leq j < n} x_{r_j, j}.$$



The construction **never** calls  $F_k()$ !

## PROOF I

We construct an adversary  $\mathcal{B}$  against the IHwPRF security of  $F_k$  using a OWF adversary  $\mathcal{A}$  against OWF.

1.  $\mathcal{B}$  queries the wPRF oracle  $2n$  times and obtains.

$$\mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \xleftarrow{\$} \mathcal{X}^{2 \times n} \text{ and } \mathbf{Y} := \begin{pmatrix} y_{0,0} & \dots & y_{0,n-1} \\ y_{1,0} & \dots & y_{1,n-1} \end{pmatrix}$$

2. It queries the oracle one more time to obtain  $x^*, y^*$ . It wishes to decide if  $y_{b,j} = F_k(x_{b,j})$  and  $y^* = F_k(x^*)$ .
3.  $\mathcal{B}$  sends  $\mathbf{X}, x^*$  to the OWF adversary  $\mathcal{A}$ .
4.  $\mathcal{A}$  outputs  $r$ , if  $x^* \neq \bigoplus_{0 \leq j < n} x_{r_j, j}$  then  $\mathcal{B}$  aborts.
5.  $\mathcal{B}$  outputs  $\bigotimes_{0 \leq j < n} y_{r_j, j} \stackrel{?}{=} y^*$ .

## PROOF II

6. If  $y^*, y_{j,b}$  were sampled uniformly at random,
  - then  $\Pr[y^* = \bigotimes_{0 \leq j < n} y_{r_j,j}] = 1/|\mathcal{Y}|$ ,
  - otherwise  $\Pr[y^* = \bigotimes_{0 \leq j < n} y_{r_j,j}] = 1$ .
7. By the LHL we know that for  $\mathbf{X} \leftarrow_{\$} \mathcal{X}^{2 \times n}$ ,  $\mathbf{r} \leftarrow_{\$} \{0, 1\}^n$ , we have that  $\oplus_{0 \leq j < n} x_{r_j,j}$  is statistically indistinguishable from  $x^*$ . Hence,  $\mathcal{B}$  correctly simulates the one-wayness game for  $\mathcal{B}$ .
8.  $\mathcal{B}$  succeeds roughly when  $\mathcal{A}$  succeeds.

# CPA-SECURE PKE I

KeyGen:

1. Sample  $k \leftarrow_{\$} \mathcal{K}$
2. Sample  $2n$  uniform elements in  $\mathcal{X}$ :

$$\mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow_{\$} \mathcal{X}^{2 \times n}$$

3. Compute

$$\mathbf{Y} := \begin{pmatrix} y_{0,0} & \dots & y_{0,n-1} \\ y_{1,0} & \dots & y_{1,n-1} \end{pmatrix} = \begin{pmatrix} F_k(x_{0,0}) & \dots & F_k(x_{0,n-1}) \\ F_k(x_{1,0}) & \dots & F_k(x_{1,n-1}) \end{pmatrix}$$

4.  $\text{sk} := k, \text{pk} := (\mathbf{X}, \mathbf{Y})$

## CPA-SECURE PKE II

Enc:

1. Sample  $r \leftarrow \{0, 1\}^n$
2. Compute  $c := \bigoplus_{0 \leq j < n} x_{r_j, j}$
3. Compute  $e := m \otimes \left( \bigotimes_{0 \leq j < n} y_{r_j, j} \right)$
4. Output  $(c, e)$

Dec:

1. Output  $m' := e \otimes F_k(c)^{-1} = m \otimes \left( \bigotimes_{0 \leq j < n} y_{r_j, j} \right) \otimes F_k \left( \bigoplus_{0 \leq j < n} x_{r_j, j} \right)^{-1}$

## PROOF I

Game <sub>0</sub>	C( $m_0, m_1$ )
1: $\text{pk}, \text{sk} \leftarrow \$ \text{KeyGen}(1^\lambda)$	1: <b>if</b> $ m_0  \neq  m_1 $ <b>then</b>
2: $b \leftarrow \$ \{0, 1\}$	2: <b>return</b> $\perp$
3: $b' \leftarrow \mathcal{D}^C(\text{pk})$	3: $c \leftarrow \$ \text{Enc}(\text{pk}, m_b)$
4: <b>return</b> $b = b'$	4: <b>return</b> $c$

The definition of IND-CPA for PKE.

## PROOF II

Game<sub>1</sub>

- 
- 1 :  $(X, Y), k \leftarrow \$ \text{KeyGen}(1^\lambda)$
  - 2 :  $b \leftarrow \$ \{0, 1\}$
  - 3 :  $b' \leftarrow \mathcal{D}^C((X, Y))$
  - 4 : **return**  $b = b'$

$C(m_0, m_1)$

- 
- 1 :  $r \leftarrow \$ \{0, 1\}^n$
  - 2 :  $c \leftarrow \bigoplus_{0 \leq j < n} X_{r_j, j}$
  - 3 :  $e \leftarrow m_b \otimes \left( \bigotimes_{0 \leq j < n} Y_{r_j, j} \right)$
  - 4 : **return**  $(c, e)$

We simply instantiated our scheme.

## PROOF III

Game <sub>2</sub>	C( $m_0, m_1$ )
1: $(X, Y), k \leftarrow \$ \text{KeyGen}(1^\lambda)$	1: $r \leftarrow \$ \{0, 1\}^n$
2: $b \leftarrow \$ \{0, 1\}$	2: $c \leftarrow \bigoplus_{0 \leq j < n} X_{r_j, j}$
3: $b' \leftarrow \mathcal{D}^C((X, Y))$	3: $e \leftarrow m_b \otimes F_k(c)$
4: <b>return</b> $b = b'$	4: <b>return</b> $(c, e)$

Correctness of IHwPRF

## PROOF IV

Game <sub>3</sub>	C( $m_0, m_1$ )
1 : $(X, Y), k \xleftarrow{\$} (\mathcal{X}^{2 \times n}, \mathcal{Y}^{2 \times n}), \mathcal{K}$	
2 : $b \xleftarrow{\$} \{0, 1\}$	1 : $r \xleftarrow{\$} \{0, 1\}^n$
3 : $b' \leftarrow \mathcal{D}^C((X, Y))$	2 : $c \leftarrow \bigoplus_{0 \leq j < n} x_{r_j, j}$
4 : <b>return</b> $b = b'$	3 : $e \leftarrow m_b \otimes F_k(c)$
	4 : <b>return</b> $(c, e)$

wPRF Security of  $F_k()$

## PROOF V

Game <sub>4</sub>	C( $m_0, m_1$ )
1: $(X, Y), k \leftarrow \$_{(\mathcal{X}^{2 \times n}, \mathcal{Y}^{2 \times n}), \mathcal{K}}$	1: $r \leftarrow \$_{\{0, 1\}^n}$
2: $b \leftarrow \$_{\{0, 1\}}$	2: $c \leftarrow \$_{\mathcal{X}}$
3: $b' \leftarrow \mathcal{D}^C((X, Y))$	3: $e \leftarrow m_b \otimes F_k(c)$
4: <b>return</b> $b = b'$	4: <b>return</b> $(c, e)$

LHL

## PROOF VI

Game <sub>4</sub>	C( $m_0, m_1$ )
1: $(X, Y), k \xleftarrow{\$} (\mathcal{X}^{2 \times n}, \mathcal{Y}^{2 \times n}), \mathcal{K}$	1: $r \xleftarrow{\$} \{0, 1\}^n$
2: $b \xleftarrow{\$} \{0, 1\}$	2: $c \xleftarrow{\$} \mathcal{X}$
3: $b' \leftarrow \mathcal{D}^C((X, Y))$	3: $e \xleftarrow{\$} \mathcal{Y}$
4: <b>return</b> $b = b'$	4: <b>return</b> $(c, e)$

wPRF security if  $F_k()$

## DH INSTANTIATION I

KeyGen:

1. Sample  $k \leftarrow \mathcal{K}$
2. Sample  $2n$  uniform elements in  $\mathcal{X}$ :

$$\mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow \mathcal{X}^{2 \times n}$$

3. Compute

$$\mathbf{Y} := \begin{pmatrix} F_k(x_{0,0}) & \dots & F_k(x_{0,n-1}) \\ F_k(x_{1,0}) & \dots & F_k(x_{1,n-1}) \end{pmatrix}$$

4.  $\text{sk} := k, \text{pk} := (\mathbf{X}, \mathbf{Y})$

KeyGen:

1. Sample  $k \leftarrow \mathbb{Z}_{p-1}$
2. Sample  $2n$  uniform elements in  $\mathbb{Z}_p^*$ :

$$\mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow \mathbb{Z}_p^{2 \times n}$$

3. Compute

$$\mathbf{Y} := \begin{pmatrix} x_{0,0}^k & \dots & x_{0,n-1}^k \\ x_{1,0}^k & \dots & x_{1,n-1}^k \end{pmatrix}$$

4.  $\text{sk} := k, \text{pk} := (\mathbf{X}, \mathbf{Y})$

## DH INSTANTIATION II

Enc:

1. Sample  $\mathbf{r} \leftarrow \{0,1\}^n$
2. Compute  $c := \bigoplus_{0 \leq j < n} x_{r_j,j}$
3. Compute  $e := m \otimes \left( \bigotimes_{0 \leq j < n} y_{r_j,j} \right)$
4. Output  $(c, e)$

Dec:

1. Output  $m' := e \otimes F_k(c)^{-1}$

Enc:

1. Sample  $\mathbf{r} \leftarrow \{0,1\}^n$
2. Compute  $c := \prod_{0 \leq j < n} x_{r_j,j}$
3. Compute  $e := m \cdot \left( \prod_{0 \leq j < n} y_{r_j,j} \right)$
4. Output  $(c, e)$

Dec:

1. Output  $m' := e \cdot c^{-k}$

Taher ElGamal. **A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.** In: CRYPTO'84. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Berlin, Heidelberg, Aug. 1984, pp. 10–18. DOI: [10.1007/3-540-39568-7\\_2](https://doi.org/10.1007/3-540-39568-7_2)

# ELGAMAL II

## KeyGen:

1. Sample  $k \leftarrow \mathbb{Z}_{p-1}$
2. Sample  $\mathbf{X} \leftarrow \mathbb{Z}_p^{2 \times n}$
3. Compute  $\mathbf{Y} := \{x_{b,j}^k\}_{b \in \{0,1\}, 0 \leq j < n}$
4.  $\text{sk} := k, \text{pk} := (\mathbf{X}, \mathbf{Y})$

## Enc:

1. Sample  $\mathbf{r} \leftarrow \{0,1\}^n$
2. Compute  $c := \prod_{0 \leq j < n} x_{r_j, j}$
3. Compute  $e := m \cdot \left( \prod_{0 \leq j < n} y_{r_j, j} \right)$
4. Output  $(c, e)$

## Dec:

1. Output  $m' := e \cdot c^{-k}$

## KeyGen:

1. Sample  $k \leftarrow \mathbb{Z}_{p-1}$
2.  $x = g$
3. Compute  $y = g^k$
4.  $\text{sk} := k, \text{pk} := (x, y)$

## Enc:

1. Sample  $r \leftarrow \mathbb{Z}_{p-1}$
2. Compute  $c := g^r$
3. Compute  $e := m \cdot y^r$
4. Output  $(c, e)$

## Dec:

1. Output  $m' := e \cdot c^{-k}$

Oded Regev. **On lattices, learning with errors, random linear codes, and cryptography.** In: *Journal of the ACM* 56.6 (Sept. 2009), 34:1–34:40. ISSN: 0004-5411 (print), 1557-735X (electronic). DOI: <http://doi.acm.org/10.1145/1568318.1568324>

Ron Rothblum. **Homomorphic Encryption: From Private-Key to Public-Key.** In: *TCC 2011*. Ed. by Yuval Ishai. Vol. 6597. LNCS. Springer, Berlin, Heidelberg, Mar. 2011, pp. 219–234. DOI: [10.1007/978-3-642-19571-6\\_14](https://doi.org/10.1007/978-3-642-19571-6_14)

## REGEV ENCRYPTION II

KeyGen:

1. Sample  $k \leftarrow \mathcal{K}$
2. Sample  $2n$  uniform elements in  $\mathcal{X}$ :

$$\mathbf{X} := \begin{pmatrix} x_{0,0} & \dots & x_{0,n-1} \\ x_{1,0} & \dots & x_{1,n-1} \end{pmatrix} \leftarrow \mathcal{X}^{2 \times n}$$

3. Compute

$$\mathbf{Y} := \begin{pmatrix} F_k(x_{0,0}) & \dots & F_k(x_{0,n-1}) \\ F_k(x_{1,0}) & \dots & F_k(x_{1,n-1}) \end{pmatrix}$$

4.  $\text{sk} := k, \text{pk} := (\mathbf{X}, \mathbf{Y})$

KeyGen:

1. Sample  $\mathbf{s} \leftarrow \mathbb{Z}_q^n$
2. Sample  $\mathbf{A} := \mathbf{A} \leftarrow \mathbb{Z}_q^{(2n \log q) \times n}$
3. Sample  $\mathbf{z} \leftarrow \chi^{2n \log q}$  (can use wPRF construction as above).
4. Compute  $\mathbf{Y} := \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{z}$
5.  $\text{sk} := \mathbf{s}, \text{pk} := (\mathbf{A}, \mathbf{b})$

## REGEV ENCRYPTION III

Enc:

1. Sample  $\mathbf{r} \leftarrow \{0,1\}^n$
2. Compute  $c := \bigoplus_{0 \leq j < n} x_{r_j,j}$
3. Compute  $e := m \otimes \left( \bigotimes_{0 \leq j < n} y_{r_j,j} \right)$
4. Output  $(c, e)$

Dec:

1. Output  $m' := e \otimes F_k(c)^{-1}$

Enc:

1. Sample  $\mathbf{r} \leftarrow \{0,1\}^{2n \log q}$
2. Compute  $\mathbf{c} := \mathbf{r}^T \cdot \mathbf{A}$
3. Compute  $e := \lfloor \frac{q}{2} \rfloor \cdot m + \mathbf{r}^T \cdot \mathbf{b}$
4. Output  $(\mathbf{c}, e)$

Dec:

1. Output  $\mathcal{R}(e - \langle \mathbf{c}, \mathbf{s} \rangle)$

# DRAGONS!

Game <sub>3</sub>	$C(m_0, m_1)$
$(X, Y), k \leftarrow \$ \text{KeyGen}()$	$r \leftarrow \$ \{0, 1\}^n$
$b \leftarrow \$ \{0, 1\}$	$c \leftarrow \bigoplus_{0 \leq j < n} X_{r_j, j}$
$b' \leftarrow \mathcal{D}^c$	$e \leftarrow m_b \otimes F_k(c)$
<b>return</b> $b = b'$	<b>return</b> $(c, e)$

Game <sub>3</sub>	$C(m_0, m_1)$
$(A, b), s \leftarrow \$ \text{KeyGen}()$	$r \leftarrow \$ \{0, 1\}^{2n \log q}$
$b \leftarrow \$ \{0, 1\}$	$c \leftarrow r^T \cdot A$
$b' \leftarrow \mathcal{D}^c$	$e \leftarrow \lfloor \frac{q}{2} \rfloor \cdot m + r^T \cdot b$
<b>return</b> $b = b'$	<b>return</b> $(c, e)$

- We cannot rely on correctness here because

$$\begin{aligned}
 F_{(k,s)}(a_0) + F_{(k,s)}(a_1) &= \langle a_0, s \rangle + D_\chi(F'_k(a_0)) + \langle a_1, s \rangle + D_\chi(F'_k(a_1)) \\
 &\neq \langle a_0 + a_1, s \rangle + D_\chi(F'_k(a_0 + a_1)) = F_{(k,s)}(a_0 + a_1)
 \end{aligned}$$

- We only have  $\mathcal{R}(F_{(k,s)}(a_0) + F_{(k,s)}(a_1)) = \mathcal{R}(F_{(k,s)}(a_0 + a_1))$ ; could output  $\mathcal{R}(e)$  instead.
- Regev solved this directly by appealing to the LHL on  $(A, b) \in \mathbb{Z}_q^{(2n \log q) \times (n+1)}$

Richard Lindner and Chris Peikert. **Better Key Sizes (and Attacks) for LWE-Based Encryption.** In: *CT-RSA 2011*. Ed. by Aggelos Kiayias. Vol. 6558. LNCS. Springer, Berlin, Heidelberg, Feb. 2011, pp. 319–339. DOI: [10.1007/978-3-642-19074-2\\_21](https://doi.org/10.1007/978-3-642-19074-2_21)

Key Idea: replace LHL with applying LWE again.

# LP11 ENCRYPTION II

## KeyGen:

1. Sample  $\mathbf{s} \leftarrow \mathbb{Z}_q^k$
2. Sample  $\mathbf{X} := \mathbf{A} \leftarrow \mathbb{Z}_q^{(2n \log q) \times n}$
3. Sample  $\mathbf{e} \leftarrow \chi^{2n \log q}$ .
4. Compute  $\mathbf{Y} := \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$
5.  $\text{sk} := \mathbf{s}$ ,  $\text{pk} := (\mathbf{A}, \mathbf{b})$

## Enc:

1. Sample  $\mathbf{r} \leftarrow \{0, 1\}^{2n \log q}$
2. Compute  $\mathbf{c} := \mathbf{r}^T \cdot \mathbf{A}$
3. Compute  $e := \lfloor \frac{q}{2} \rfloor \cdot m + \mathbf{r}^T \cdot \mathbf{b}$
4. Output  $(\mathbf{c}, e)$

## Dec:

1. Output  $\mathcal{R}(e - \langle \mathbf{c}, \mathbf{s} \rangle)$

## KeyGen:

1. Sample  $\mathbf{s} \leftarrow \mathbb{Z}_q^k$
2. Sample  $\mathbf{X} := \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times n}$
3. Sample  $\mathbf{e} \leftarrow \chi^n$ .
4. Compute  $\mathbf{Y} := \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$
5.  $\text{sk} := \mathbf{s}$ ,  $\text{pk} := (\mathbf{A}, \mathbf{b})$

## Enc:

1. Sample  $\mathbf{r}, \mathbf{d}, d' \leftarrow \chi^n, \chi^n, \chi$
2. Compute  $\mathbf{c} := \mathbf{r}^T \cdot \mathbf{A} + \mathbf{d}$
3. Compute  $e := \lfloor \frac{q}{2} \rfloor \cdot m + \mathbf{r}^T \cdot \mathbf{b} + d'$
4. Output  $(\mathbf{c}, e)$

## Dec:

1. Output  $\mathcal{R}(e - \langle \mathbf{c}, \mathbf{s} \rangle)$

FIN

READ MINICRYPT PRIMITIVES WITH ALGEBRAIC STRUCTURE AND  
APPLICATIONS!

## REFERENCES I

- [AMPR23] Navid Alamati, Hart Montgomery, Sikhar Patranabis, and Arnab Roy. **Minicrypt Primitives with Algebraic Structure and Applications.** In: *Journal of Cryptology* 36.1 (Jan. 2023), p. 2. DOI: [10.1007/s00145-022-09442-2](https://doi.org/10.1007/s00145-022-09442-2).
- [BM84] Manuel Blum and Silvio Micali. **How to Generate Cryptographically Strong Sequences of Pseudorandom Bits.** In: *SIAM Journal on Computing* 13.4 (1984), pp. 850–864.
- [ElG84] Taher ElGamal. **A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.** In: *CRYPTO'84*. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Berlin, Heidelberg, Aug. 1984, pp. 10–18. DOI: [10.1007/3-540-39568-7\\_2](https://doi.org/10.1007/3-540-39568-7_2).

## REFERENCES II

- [GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. **How to Construct Random Functions.** In: *Journal of the ACM* 33.4 (Oct. 1986), pp. 792–807. DOI: [10.1145/6490.6503](https://doi.org/10.1145/6490.6503).
- [Lam79] Leslie Lamport. **Constructing Digital Signatures from a One-way Function.** Technical Report SRI-CSL-98. SRI International Computer Science Laboratory, Oct. 1979.
- [LP11] Richard Lindner and Chris Peikert. **Better Key Sizes (and Attacks) for LWE-Based Encryption.** In: *CT-RSA 2011*. Ed. by Aggelos Kiayias. Vol. 6558. LNCS. Springer, Berlin, Heidelberg, Feb. 2011, pp. 319–339. DOI: [10.1007/978-3-642-19074-2\\_21](https://doi.org/10.1007/978-3-642-19074-2_21).
- [LR88] Michael Luby and Charles Rackoff. **How to construct pseudorandom permutations from pseudorandom functions.** In: *SIAM Journal on Computing* 17.2 (1988).

## REFERENCES III

- [Mer79] Ralph Charles Merkle. **Secrecy, authentication, and public key systems.** PhD thesis. Stanford university, 1979.
- [NY89] Moni Naor and Moti Yung. **Universal One-Way Hash Functions and their Cryptographic Applications.** In: *21st ACM STOC*. ACM Press, May 1989, pp. 33–43. DOI: [10.1145/73007.73011](https://doi.org/10.1145/73007.73011).
- [Reg09] Oded Regev. **On lattices, learning with errors, random linear codes, and cryptography.** In: *Journal of the ACM* 56.6 (Sept. 2009), 34:1–34:40. ISSN: 0004-5411 (print), 1557-735X (electronic). DOI: <http://doi.acm.org/10.1145/1568318.1568324>.
- [Rot11] Ron Rothblum. **Homomorphic Encryption: From Private-Key to Public-Key.** In: *TCC 2011*. Ed. by Yuval Ishai. Vol. 6597. LNCS. Springer, Berlin, Heidelberg, Mar. 2011, pp. 219–234. DOI: [10.1007/978-3-642-19571-6\\_14](https://doi.org/10.1007/978-3-642-19571-6_14).

- [Sch90] Claus-Peter Schnorr. **Efficient Identification and Signatures for Smart Cards.** In: CRYPTO'89. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, New York, Aug. 1990, pp. 239–252. DOI: [10.1007/0-387-34805-0\\_22](https://doi.org/10.1007/0-387-34805-0_22).