算術回路に対するGKW変換の検討

知念 広太郎[1,a)]　穴田 啓晃[2,b)]

概要：Goyal，Koppula，Waters が TCC2016-B で発表した，選択的安全な関数型暗号から準適応的安全な関数型暗号への変換が知られている（GKW 変換）．この変換にはブール回路の garble 化が用いられている．本稿では，Applebaum，Ishai，Kushilevitz が FOCS2011 で発表した算術回路の garble 化の手法を用いることで，算術回路に対する GKW 変換を検討する．

キーワード：関数型暗号,Garbled ciruit, 算術回路, 準適応的安全性

Towards GKW Transformation for Arithmetic Circuits

Kotaro Chinen[1,a)]　Hiroaki Anada[2,b)]

Abstract: There is a transformation of a selectively secure functional encryption scheme into semi-adaptively secure functional encryption scheme, which was proposed at TCC2016-B by Goyal, Koppula and Waters (the GKW transformation). In the transformation, the garbling technique for a boolean circuit is used. In this paper, we try to construct a GKW transformation for an arithmetic circuit by using the garbling technique for an arithmetic circuit proposed by Applebaum, Ishai and Kushilevitz.

Keywords: functional encryption,Garbled Circuits,Arithmetic Circuits ,semi-adaptively secure

1. 研究の背景

Goyal,Koppula,Waters が TCC2016-B で，選択的安全な関数型暗号を公開鍵暗号方式と論理回路に対する Garbled Circuits を用いることでより安全性の強い準適応的安全な関数型暗号へ変換する方法 (GKW 変換) を示した．オリジナルの GKW 変換では，選択的安全な暗号スキームの暗号化アルゴリズムを論理回路で設計し，Garbled Circuits に変換する手法をとっている．そのため選択的安全な暗号スキームの公開鍵の bit 長を参照する必要があり，結果として，変換により得られる準適応的安全な暗号スキームは暗号文長，秘密鍵長，公開鍵長が大きくなるという課題がある．

[1] 長崎県立大学 地域創生研究科　情報工学専攻
,University of Nagasaki
[2] 長崎県立大学
,University of Nagasaki
[a)] mc120002@sun.ac.jp
[b)] anada@sun.ac.jp

ビットごとに処理する論理回路に対し，有限体 F_q 上の成分ごとに処理する算術回路と呼ばれる回路設計が知られている．算術回路は F_q 上の成分ごとの処理のため計算効率の向上が期待でき，また，データ長を短くできるといった利点も挙げられる．このことから，オリジナルの GKW 変換の課題を解決することが可能ではないかと考えている．

そこで，我々は選択的安全な暗号スキームの暗号化アルゴリズムを論理回路ではなく算術回路で設計する GKW 変換を考える．GKW 変換は，選択的安全な暗号スキームの調整を必要とすることなく，準適応的安全な暗号に引き上げることができる．そのため，オリジナルの GKW 変換の課題を解決することは，利便性の向上につながると考える．

1.1 我々の貢献

本稿では，暗号化アルゴリズムを算術回路で設計した場合の GKW 変換の 2 つの検討案を示す．算術回路に対する Garbled Circuits では，算術回路の各入力ワイヤに対して，関数 f を割り当て，入力 $x \in F_q$ が与えられた時，$f(x)$ を

ワイヤ鍵とする．

初めに，オリジナルの GKW 変換と同様に，入力ワイヤに割り当てられた関数の入力 $x \in F_q$ 全てのパターンに対応した q 個のワイヤ鍵を生成し，暗号化して暗号文の要素としえて使用する検討案の一般的な構成法を示す．また，関数型暗号が選択的安全であり，公開鍵暗号方式が Semantically Secure，算術回路に対する Garbled Circuits が安全なスキームを満たすならば，その検討案の変換により得られる関数型暗号が準適応的安全性であることを示す．

次に，ワイヤ鍵を生成するのではなく，関数 f を暗号化して暗号文の要素に検討案の一般的な構成法を示す．また，安全性証明の道筋と解決すべき点を示す．

2. 準備

本節では，用語についての解説を示す．$\mathbb{N}$ は自然数の集合である．λ はセキュリティパラメータで，$\lambda \in \mathbb{N}$ である．F_q は素数 q の有限体である．$A = (x_1, \cdots, x_e)$ の時，A の i 番目の要素は $A[x_i]$ と表記する．

2.1 公開鍵暗号 (PKE)

本節では，公開鍵暗号の構成を示す．メッセージ空間 M_λ に対する公開鍵暗号スキームは 3 つのアルゴリズム $(\mathsf{Setup_{PKE}}, \mathsf{Enc_{PKE}}, \mathsf{Dec_{PKE}})$ で構成される．

$\mathsf{Setup_{PKE}}(1^\lambda) \to (\mathrm{PK}, \mathrm{SK})$. $\mathsf{Setup_{PKE}}$ アルゴリズムは，引数にセキュリティパラメータ 1^λ を受け取り，公開鍵 PK と秘密鍵 SK を出力する．

$\mathsf{Enc_{PKE}}(\mathrm{PK}, m \in M_\lambda) \to \mathsf{c}$. $\mathsf{Enc_{PKE}}$ アルゴリズムは，引数に公開鍵 $\mathrm{PK_{PKE}}$ とメッセージ空間 M_λ 上のメッセージ m を受け取り，暗号文 c を出力する．

$\mathsf{Dec_{PKE}}(\mathrm{SK}, \mathsf{c}) \to M_\lambda$. $\mathsf{Dec_{PKE}}$ アルゴリズムは，引数に秘密鍵 $\mathrm{SK_{PKE}}$ と暗号文 c を受け取り，メッセージ m を出力する．

2.1.1 公開鍵暗号の安全性定義

本節では，公開鍵暗号に対して以下の実験を定義する．

$$
\begin{aligned}
&Expr^{\mathrm{SS}}_{\mathcal{A},\mathsf{PKE}}(\lambda, n):\\
&\quad (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Setup}(1^\lambda)\\
&\quad (m_0^*, m_1^*) \leftarrow \mathcal{A}(\mathrm{PK}, 1^\lambda)\\
&\quad b \leftarrow \{0,1\}, ct^* \leftarrow \mathsf{Enc}(\mathrm{PK}, m_b^*)\\
&\quad b' \leftarrow \mathcal{A}(ct^*)\\
&\quad \mathsf{If} b' = b \ \mathsf{then\ Return\ } \textsc{Win} \ \mathsf{else\ Return\ } \textsc{Lose}
\end{aligned}
$$

実験において，攻撃者 $\mathcal{A}$ は入力に $(\mathrm{MPK}, 1^\lambda)$ を受け取り，チャレンジメッセージ (m_0^*, m_1^*) を出力する．

2.2 関数型暗号 (FE) [3]

本節では，関数型暗号の構成を示す．メッセージ空間 M_n 及び関数空間 F_n の関数型暗号スキームは 4 つの多項式時間アルゴリズム $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ で構成される．

$\mathsf{Setup}(1^\lambda, 1^n) \to (\mathrm{MPK}, \mathrm{MSK})$. Setup アルゴリズムは，引数にセキュリティパラメータ 1^λ と機能指数 1^n を受け取り，マスタ公開鍵/マスタ秘密鍵 $(\mathrm{MPK}, \mathrm{MSK})$ のペアを出力する．

$\mathsf{Enc}(\mathrm{MPK}, m \in M_n) \to \mathrm{CT}$. Enc アルゴリズムは，引数にマスタ公開鍵 MPK とメッセージ $m \in M_N$ を受け取り，暗号文 CT を出力する．

$\mathsf{KeyGen}(\mathrm{MSK}, f \in F) \to \mathrm{SK}_f$. KeyGen アルゴリズムは，引数にマスタ秘密鍵 MSK と関数 $f \in F$ を受け取り，秘密鍵 SK_f を出力する．

$\mathsf{Dec}(\mathrm{SK}_f, \mathrm{CT}) \to \{0, 1, \perp\}$ Dec アルゴリズムは，引数に秘密鍵 SK_f と暗号文 CT を受け取り，$y \in \{0, 1, \perp\}$ を出力する．

2.2.1 関数型暗号の選択的安全性 [4] と準適応的安全性 [2]

本節では，関数型暗号に対して以下のような実験を定義する．

$$
\begin{aligned}
&\mathsf{Exp}^{\text{Ind-x-cpa}}_{\mathcal{A},FE}(1^\lambda, 1^n):\\
&\quad (\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)\\
&\quad (m_0^*, m_1^*) \leftarrow \mathcal{A}(x, 1^n)\\
&\quad b \leftarrow \{0,1\}, ct^* \leftarrow \mathsf{Enc}(\mathrm{MPK}, m_b^*)\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{KG}(msk,\cdot)}(\mathrm{MPK}, ct^*)\\
&\quad \mathsf{If} b' = b \ \mathsf{then\ Return\ } \textsc{Win} \ \mathsf{else\ Return\ } \textsc{Lose}
\end{aligned}
$$

$x = 1^\lambda$: 選択的安全性　$x = \mathrm{MPK}$: 準適応的安全性

実験において，攻撃者 $\mathcal{A}$ は入力に $(x, 1^n)$ を受け取り，チャレンジメッセージ (m_0^*, m_1^*) を出力する．もし $\mathcal{A}$ が受け取る入力が $(1^\lambda, 1^n)$ である場合，選択的な攻撃であるという．もし $\mathcal{A}$ が受け取る入力が $(\mathrm{PK}, 1^n)$ である場合，準適応的な攻撃であるという．$\mathcal{A}$ は鍵生成オラクル $\mathcal{KG}$ へアクセスすることが可能である．$\mathcal{KG}$ は関数 $f \in F_n$ でクエリを受け取ると，秘密鍵 SK_f を返す．

2.3 The Lerning with error(LWE) 問題 [5]

LWE 問題とは計算困難問題の一種である．セキュリティパラメータ λ とした時，次元 $k = k(\lambda)$，法 $q = q(\lambda)$，ノイズ分布 $\chi = \chi(\lambda)$，情報分布 $S = S(\lambda)$ が与えられる．χ, S は $\mathbb{Z}_{q(\lambda)}$ 上の確率分布のペアとする．U_q は $\mathbb{Z}_{q(\lambda)}$ 上の一様分布とする．決定的 LWE 問題 $\mathsf{LWE}(k, q, \chi, S)$ は以下の場合，$(t(\lambda), \epsilon(\lambda))$-計算困難である．

$\mathsf{LWE}(k, q, \chi, S)$ の分布が，：

・$(M, r = Ms + e | M \xleftarrow{R} U_q^{tk}, s \xleftarrow{R} S^k, e \xleftarrow{R} \chi^t)$

一様分布 (U_q^{tk}, U_q^t) と (t, ϵ)-識別不可能である．

LWE の仮定は，上記が $t = \lambda^{\omega(1)}$ および $\epsilon = \lambda^{-\omega(1)}$ で成り立つことを主張する．

2.4 算術回路に対する Garbled circuits [1]

本節では, 算術回路に対する Garbled Circuits の構成方法を示す．先行研究として，Applebaum,Ishai,Kushilevitz [1] が紹介した方式を適用する．この方式では，算術回路に Decomposable affine randomized encoding(DARE) を適用することにより Garbled Circuits を実現する．算術回路をランダム化された分解可能アフィン回路に変換するする DARE compiler は，以下の入力と出力を持つ多項式時間アルゴリズムである．

・Input:セキュリティパラメータ 1^λ，算術回路 C，回路上のワイヤの値を抑える正の整数 U

・Output:Encd(エンコーダ)，Decd(デコーダ)，Sim(シミュレータ)．通常，Encd は，入力 $x = (x_1, \cdots, x_e)$ を受け取ると，出力 $(W, L_1, \cdots, L_e)$ を出力する．行列 W(Grbled Circuits に相当)，アフィン鍵 L_i(ワイヤ鍵に相当) はアフィン関数 $L_i(\cdot)$ に入力 x_i を適用し計算することで得る．Encd を構築には，Affinization gadget と Key shrinking gadget を使用する．Affinization gadget は統計的に安全性が示され，Key shrinking gadget は LWE 仮定に基づき安全性が示されている．Applebaum らの手法で使用される LWE 仮定の法 q は，$q = 2^{(k^{1/\gamma})} : (\gamma > 1)$ である．

2.4.1 DARE compiler の安全性定義

本節では，DARE compiler に対して以下のような実験を定義する．

$$
\begin{aligned}
&Expr^{\mathrm{Privacy}}_{\mathcal{A},\mathsf{DARE\ compiler}}(\lambda):\\
&\quad \mathsf{C}, U, \leftarrow \mathcal{A}(1^\lambda)\\
&\quad (\mathsf{Encd}, \mathsf{Decd}, \mathsf{Sim}) \xleftarrow{R} T(1^\lambda, \mathsf{C}, U)\\
&\quad b \leftarrow \{0,1\}\\
&\quad \text{if } b = 0:\\
&\quad b' \leftarrow \mathcal{A}(\mathsf{Encd}(x : r_e))\\
&\quad \text{if } b = 1:\\
&\quad b' \leftarrow \mathcal{A}(\mathsf{Sim}(\mathsf{C}(x) : r_s))\\
&\quad \mathsf{If} b' = b \text{ then Return } \textsc{Win} \text{ else Return } \textsc{Lose}
\end{aligned}
$$

3. 算術回路に対する GKW 変換

本節では, 暗号化アルゴリズムを算術回路で設計した場合の選択的安全な関数型暗号から準適応的安全な関数型暗号への一般的変換の検討案を示す．

3.1 検討案 1

3.1.1 一般的な構成

変換により得られる関数型暗号は 4 つの多項式時間アルゴリズム (Setup,Enc,KeyGen,Dec) で構成される．

$\mathsf{Setup}(1^\lambda, 1^n) \rightarrow (\mathrm{PK}, \mathrm{MSK})$．Setup アルゴリズムはセキュリティパラメータ 1^λ, 機能指数 1^n を受け取る．初めに，Setup_{sel} を実行し，鍵セット $(\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel})$ を得る．$\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}$ は，e 個の有限体 F_q 上の要素で構成されたベクトルである．次に，$\mathsf{Setup}_{\mathsf{PKE}}$ を実行し，eq 個の鍵セット $(\mathrm{PK}_{i,j}, \mathrm{SK}_{i,j})_{i \le e, j \le q}$ を得る．最後に，マスタ公開鍵 $\mathrm{MPK} = (\mathrm{PK}_{i,j})$ とマスタ秘密鍵 $\mathrm{MSK} = (\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}, \mathrm{SK}_{i,j})$ を出力する．

$\mathsf{Enc}(\mathrm{MPK}, m \in M_n) \rightarrow \mathrm{CT}$. Enc アルゴリズムは，マスタ公開鍵 MSK とメッセージ空間 M_n 上の平文 m を受け取る．初めに，選択的安全な関数型暗号の暗号化アルゴリズム Enc_{sel} に平文 m を入力し，算術回路 $\mathsf{Enc}_{sel}(\cdot, m : r)$ を計算する．次に，DARE compiler に算術回路 $\mathsf{Enc}_{sel}(\cdot, m : r)$ とセキュリティパラメータ 1^λ，算術回路の各ワイヤの取りうる上限値 U を入力し，エンコーダ Encd とデコーダ Decd を得る．この時，Encd は，行列 W と e 個のアフィン関数 $L_i(x)$ で構成されている．次に，各アフィン関数に有限体 F_q 上の各値を代入し，eq 個のアフィン鍵を得る．その後，$\mathsf{Enc}_{\mathsf{PKE}}$ を実行し，各アフィン鍵を対応する公開鍵 $\mathrm{PK}_{i,j}$ で暗号化し，$\mathsf{c}_{i,j}$ を得る．最後に，暗号文 $\mathrm{CT} = \{\mathsf{Decd}, W, \{\mathsf{c}_{i,j}\}_{i \le e, j \le q}\}$ を出力する．

$\mathsf{KeyGen}(\mathrm{MSK}, f \in F_n) \rightarrow \mathrm{SK}_f$. KeyGen アルゴリズムは，マスタ秘密鍵 MSK と関数空間 F_n 上の関数 f を受け取る．初めに，KeyGen_{sel} アルゴリズムを実行し，$\mathrm{SK}_{f,sel}$ を得る．次に，mq 個の PKE の秘密鍵から MPK_{sel} の要素の値に対応する m 個の秘密鍵を選択する．最後に，秘密鍵 $\mathrm{SK} = (\mathrm{MPK}_{sel}, \mathrm{SK}_{f,sel}, \mathrm{SK}_{i,\mathrm{MPK}_{sel}[x_i]})$ を出力する．

$\mathsf{Dec}(\mathrm{SK}_{f,sel}, \mathrm{CT}) \rightarrow \{0, 1, \perp\}$. Dec アルゴリズムは，秘密鍵 SK_f と暗号文 CT を受け取る．初めに，$\mathsf{Dec}_{\mathsf{PKE}}$ を実行し，$\mathsf{c}_{i,j}$ を復号し，e 個のアフィン鍵 $L_i(\mathrm{MPK}_{sel}[x_i])$ を得る．次に，Decd に W と $L_i(\mathrm{MPK}_{sel}[x_i])$ を入力し，暗号文 $\tilde{\mathrm{C}}\mathrm{T}$ を得る．最後に，Dec_{sel} を実行し，暗号文 $\tilde{\mathrm{C}}\mathrm{T}$ を復号した平文 $\tilde{m}$ を求める．

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

$(\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}) \leftarrow \mathsf{Setup}_{sel}(1^\lambda, 1^n)$
for $i = 1$ to e **do**
  for $j = 1$ to q **do**
    $(\mathrm{PK}_{i,j}, \mathrm{SK}_{i,j}) \leftarrow \mathsf{Setup}_{\mathsf{PKE}}(1^\lambda)$
  end for
end for
$\mathrm{MPK} = \{\mathrm{PK}_{i,j}\}_{i<=e, j<=q}$
$\mathrm{MSK} = \{\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}, \{\mathrm{SK}_{i,j}\}_{i \le e, j \le q}\}$
return $(\mathrm{MPK}, \mathrm{MSK})$

Algorithm 2 $\mathsf{Enc}(\mathrm{MPK}, m \in M_n)$

$ckt = \mathsf{Enc}_{sel}(x, m : r)$
$(\mathsf{Encd}, \mathsf{Decd}) \leftarrow \mathsf{DARE\ compiler}(ckt, 1^\lambda, U)$
for $i = 1$ to m **do**
  for $j = 1$ to q **do**
    $\mathsf{c}_{i,j} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}_{i,j}, L_i(j-1))$
  end for
end for
$\mathrm{CT} = \{\mathsf{Decd}, W, \{\mathsf{c}_{i,j}\}_{i \le m, j \le q}\}$
return CT

Algorithm 3 $\mathsf{KeyGen}(\mathrm{MSK}, f \in F_n)$

$\mathrm{SK}_{sel,f} \leftarrow \mathsf{KeyGen}_{sel}(\mathrm{MSK}_{sel}, f)$
$\mathrm{SK}_f = \{\mathrm{MPK}_{sel}, \mathrm{SK}_{sel,f}, \mathrm{SK}_{i,\mathrm{MPK}_{sel}[x_i]}\}$
return SK_f

Algorithm 4 $\mathsf{Dec}(\mathrm{SK}_f, \mathrm{CT})$

for $i = 0$ to m **do**
  $L_i(\mathrm{MPK}_{sel}[x_i]) \leftarrow \mathsf{Dec}_{\mathsf{PKE}}(\mathrm{SK}_{i,\mathrm{MPK}_{sel}[x_i]}, \mathsf{c}_{i,\mathrm{MPK}_{sel}[x_i]})$
end for
$\tilde{\mathrm{CT}} \leftarrow \mathsf{Decd}(W, \{L_i(\mathrm{MPK}_{sel}[x_i])\}_{i \le m})$
$\tilde{m} \leftarrow \mathsf{Dec}_{sel}(\mathrm{SK}_{f,sel}, \tilde{\mathrm{CT}})$
return $y \in \{0, 1, \perp\}$

3.1.2 安全性証明

Theorem 1. メッセージ空間 M_n，関数空間 F_n に対する $FE_{sel} = (\mathsf{Setup}_{sel}, \mathsf{Enc}_{sel}, \mathsf{KeyGen}_{sel}, \mathsf{Dec}_{sel})$ が選択的安全な関数型暗号であり，*DARE compiler* により得られる *Garbled Circuit* が安全なスキームであり，$PKE = \mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Enc}_{\mathsf{PKE}}, \mathsf{Dec}_{\mathsf{PKE}}$ が *Semantically Secure* な公開鍵暗号方式である時，変換により得られる FE は同様の M_n, F_n に対して，準適応的安全な関数型暗号である．

$$Adv^{semi-adapt}_{\mathcal{A},FE}(1^\lambda, 1^n) \le Adv^{sel}_{\mathcal{A},FE_{sel}}(1^\lambda, 1^n) + Adv^{Privacy}_{\mathcal{A},\mathsf{DARE\ compiler}}(1^\lambda, 1^n) + Adv^{SS}_{\mathcal{A},PKE}(1^\lambda, 1^n)$$

3.1.3 ゲーム

安全性証明を示すために，以下の 3 つのゲームを考える．Game 1 は準適応的安全性ゲームである．

*Game*1 :

1.(Setup Pahse)

$\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel} \leftarrow \mathsf{Setup}_{sel}(1^\lambda, 1^n)$

$(\mathrm{PK}_{i,j}, \mathrm{SK}_{i,j})_{i \le m, j \le q} \leftarrow \mathsf{Setup}_{\mathsf{PKE}}(1^\lambda)$

$\mathrm{MPK} = \{\mathrm{PK}_{i,j}\}_{i \le m, j \le q}$

$\mathrm{MSK} = \{\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}, \{\mathrm{SK}_{i,j}\}_{i \le m, j \le q}\}$

$\mathcal{A} \leftarrow \mathrm{MPK}$

2.(Challenge Phase)

$(m^*_0, m^*_1) \leftarrow \mathcal{A}(\mathrm{MPK}, 1^n)$:

$b \leftarrow \{0, 1\}$

$(\mathsf{Encd}, \mathsf{Decd}) \leftarrow \mathsf{DARE\ compiler}(\mathsf{Enc}_{sel}(x, m^*_b; r), 1^\lambda, U)$

$\mathsf{Encd} := (W, L_i(\cdot))$

$\mathsf{c}_{i,j} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}_{i,j}, L_i(j)_{\{i \le m, j \le q\}})$

$\mathrm{CT} = \{\mathsf{Decd}, W, \{\mathsf{c}_{i,j}\}_{i \le m, j \le q}\}$

$\mathcal{A} \leftarrow \mathrm{CT}$

3.(Key Query Phase)

$f \leftarrow \mathcal{A}$,

$\mathrm{SK}_{f,sel} \leftarrow \mathcal{KG}(\mathrm{MSK}_{sel}, f)$

$\mathrm{SK}_f = (\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}, \mathrm{SK}_{i,\mathrm{MPK}_{sel}[x_i]})$

$\mathcal{A} \leftarrow \mathrm{SK}_f$

4.(Guess)

$b' \leftarrow \mathcal{A}$

if $b' = b$ then Return Wins else Retune Lose

Game 2 は，Game 1 とチャレンジ暗号文の生成方法以外は同じである．Game 1 との違いは，$j = \mathrm{MPK}_{sel}[x_i]$ の時，アフィン鍵 $L_i(j)$ を PKE で暗号化する．それ以外の場合，アフィン鍵ではなく 0 を暗号化する．

*Game*2 :

2.(Challenge Phase)

$(m^*_0, m^*_1) \leftarrow \mathcal{A}(\mathrm{MPK}, 1^n)$:

$b \leftarrow \{0, 1\}$

$(\mathsf{Encd}, \mathsf{Decd}) \leftarrow \mathsf{DARE\ compiler}(\mathsf{Enc}_{sel}(x, m^*_b; r), 1^\lambda, U)$

<u>if $j = \mathrm{MPK}_{sel}[x_i]$:</u>

<u>$\mathsf{c}_{i,j} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}_{i,j}, L_i(j))_{\{i \le m, j \le q\}}$</u>

<u>otherwise:</u>

<u>$\mathsf{c}_{i,j} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}_{i,j}, 0)_{\{i \le m, j \le q\}}$</u>

$\mathrm{CT} = \{\mathsf{Decd}, W, \{\mathsf{c}_{i,j}\}_{i \le m, j \le q}\}$

$\mathcal{A} \leftarrow \mathrm{CT}$

Game3 は，Game2 とチャレンジ暗号文の生成方法以外は同じである．Game 2 との違いは，Garbled Circuits とワイヤ鍵の生成は Sim でシミュレートされていることである．

$Game3$:

2.(Challenge Phase)

$(m_0^*, m_1^*) \leftarrow \mathcal{A}(\mathrm{MPK}, 1^n)$:

$b \leftarrow \{0,1\}$

$(\underline{\mathsf{Sim}}, \mathsf{Decd}) \leftarrow \mathsf{DARE\ compiler}(\mathsf{Enc}_{sel}(x, m_b^*; r), 1^\lambda, U)$

$\mathsf{c}^* \leftarrow \mathsf{Enc}_{sel}(\mathrm{MPK}_{sel}, m_b^*)$

$\underline{(W, L_i) \leftarrow \mathsf{Sim}(\mathsf{c}^*; r)}$

if $j = \mathrm{MPK}_{sel}[x_i]$:

$\mathsf{c}_{i,j} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}_{i,j}, L_i)_{\{i \leq m\}}$

otherwise:

$\mathsf{c}_{i,j} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}_{i,j}, 0)_{\{i \leq m, j \leq q\}}$

$\mathrm{CT} = \{\mathsf{Decd}, W, \{\mathsf{c}_{i,j}\}_{i \leq m, j \leq q}\}$

$\mathcal{A} \leftarrow \mathrm{CT}$

3.1.4 式

本説では，Theorem1 が成立することを証明するための 3 つの補題の証明を示す．

Game i における，攻撃者 $\mathcal{A}$ のアドバンテージは $Adv_{\mathcal{A}}^{gamei} = |Pr[\mathcal{A}\ \mathrm{Wins}] - 1/2|$ で定義される．

補題 1：FE に対する任意の多項式時間の準適応的 CPA 攻撃者 $\mathcal{A}$ を内部に雇うことで，PKE に対する多項式時間の Semantically Secure 攻撃者 $\mathcal{B}$ を構築できることを示す．Game1 と Game2 の識別不可能性を示すために，eq-混合 Game を考える．e は MPK_{sel} の要素の数である．q は有限体 F_q の元の要素数である．ij 番目混合 Game では，$a \leq i$ で，$b \leq j$ かつ $b \neq \mathrm{MPK}_{sel}[x_i]$ の場合はワイヤ鍵 $L_a(b-1)$ ではなく 0 の PKE 暗号化を行う．$a \geq i$ で，$b > j, b \neq \mathrm{MPK}_{sel}[x_i]$ の場合はワイヤ鍵 $L_a(b-1)$ の PKE 暗号化を行う．$i = 0, j = 0$ の時，Game1 と等しく，$i = e, j = q$ の時，Game2 に等しい．

初めに，$\mathcal{B}$ はチャレンジ公開鍵 PK^* を受け取る．次に，$\mathcal{B}$ は $\mathsf{Setup}_{\mathsf{PKE}}$ と Setup_{sel} を走らせる．この時，$j \neq \mathrm{MPK}_{sel}[x_i]$ となる公開鍵 $\mathrm{PK}_{i,j}$ の一つ $(\mathrm{PK}_{a,b})$ と PK^* を入れ替える．$\mathcal{A}$ へ MPK を送る．$\mathcal{A}$ はチャレンジメッセージ (m_0^*, m_1^*) を $\mathcal{B}$ へ送る．$\mathcal{B}$ は算術回路 C と $\mathsf{c}_{a,b}$ を除く $\mathsf{c}_{i,j}$ を作成する．$\mathcal{B}$ は自身のチャレンジメッセージ $(L_a(b),0)$ を，チャレンジャへ送る．その後，$\mathcal{B}$ は受け取ったチャレンジ暗号文を組み込み，$\mathcal{A}$ の暗号文を生成する．$\mathcal{A}$ は b' を出力し，$\mathrm{Win}(b' = b)$ の場合，$\mathcal{B}$ は 0 が暗号化されたと推測し，それ以外の場合，$L_a(b)$ が暗号化されたと推察する．$\mathcal{B}$ の内部にある $\mathcal{A}$ は，0 が暗号化された場合，ab-混合 Game の View と同じで，$L_a(b)$ が暗号化された場合，a(b-1)-混合 Game の View と同じである．それ故，$|Adv_{\mathcal{A}}^{game1} - Adv_{\mathcal{A}}^{game2}|$ が無視できないならば，PKE スキームは Semantically Secure ではない．

補題 2：FE に対する任意の多項式時間の準適応的 CPA 攻撃者 $\mathcal{A}$ を内部に雇うことで，DARE compiler に対する多項式時間の Privacy 攻撃者 $\mathcal{B}$ を構築できることを示す．

初めに，$\mathcal{B}$ は $\mathsf{Setup}_{\mathsf{PKE}}$ と Setup_{sel} を走らせる．$\mathcal{A}$ へ MPK を送る．$\mathcal{A}$ はチャレンジメッセージ (m_0^*, m_1^*) を $\mathcal{B}$ へ送る．$\mathcal{B}$ は，$b \xleftarrow{R} 0,1$ 行い，算術回路 $\mathsf{C} = \mathsf{Enc}_{sel}(\cdot, m_b^*; r)$ を計算する．$\mathcal{B}$ は $\mathsf{C}, \mathrm{MPK}_{sel}$ を，チャレンジャへ送る．その後，$\mathcal{B}$ は受け取った Garbled Circuits$W$ と m 個のアフィン鍵 Li を受け取る．$\mathcal{B}$ は，受け取ったアフィン鍵以外に必要なアフィン鍵を 0 として置き，チャレンジ暗号文を作成し，$\mathcal{A}$ へ送る．$\mathcal{A}$ は b' を出力し，$\mathrm{Win}(b' = b)$ の場合，$\mathcal{B}$ は Encd により回路が Garbled されたと推測し，それ以外の場合，Sim によりシミュレートされたと推測する．$\mathcal{B}$ の内部にある $\mathcal{A}$ は，Encd の場合，Game2 の View と同じで，Sim の場合，Game3 の View と同じである．それ故，$|Adv_{\mathcal{A}}^{game2} - Adv_{\mathcal{A}}^{game3}|$ が無視できないならば，DARE compiler は安全ではない．

補題 3：FE に対する任意の多項式時間の準適応的 CPA 攻撃者 $\mathcal{A}$ を内部に雇うことで，FE_{sel} に対する多項式時間の選択的 CPA 攻撃者 $\mathcal{B}$ を構築できることを示す．初めに，$\mathcal{B}$ は $\mathsf{Setup}_{\mathsf{PKE}}$ を走らせる．$\mathcal{A}$ へ MPK を送る．$\mathcal{A}$ はチャレンジメッセージ (m_0^*, m_1^*) を $\mathcal{B}$ へ送る．$\mathcal{B}$ は，自身のチャレンジメッセージとして (m_0^*, m_1^*) をチャレンジャへ送る．チャレンジャは $b^* \xleftarrow{R} 0,1$ を選択し，Setup_{sel} を走らせ，$\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathrm{MPK}_sel, m_b^*)$ 計算する．チャレンジャは，$\mathsf{c}^*, \mathrm{MPK}_{sel}$ を $\mathcal{B}$ へ送る．$\mathcal{B}$ は，$\hat{b} \xleftarrow{R} 0,1$ を選択し，DARE compiler で $\mathsf{Sim}_{\hat{b}}$ を生成する．$\mathcal{B}$ は，$(W, L_i) \leftarrow \mathsf{Sim}_{\hat{b}}(\mathsf{c}^*; r)$ を生成する．$\mathcal{B}$ は，残りの必要なアフィン鍵を 0 として置き，チャレンジ暗号文を作成し，$\mathcal{A}$ へ送る．この時，$\hat{b} = b^*$ ならば，$\mathcal{A}$ は b' を出力する．それ以外ならば，$\mathcal{A}$ は $\perp$ を出力する．$\mathcal{B}$ は，受け取った b' をそのまま自身の推測として出力する．$\mathcal{B}$ の内部にある $\mathcal{A}$ は，Game3 の View と同じである．それ故，$Adv_{\mathcal{A}}^{game3}$ が無視できないならば，FE_{sel} は選択的安全ではない．

補題 1,2,3 よって，Theorem1 の不等式は成り立つ．

3.2 検討案 2

変換により得られる関数型暗号は 4 つの多項式時間アルゴリズム (Setup,Enc,KeyGen,Dec) で構成される．

$\mathsf{Setup}(1^\lambda, 1^n) \rightarrow (\mathrm{PK}, \mathrm{MSK})$．Setup アルゴリズムはセキュリティパラメータ 1^λ, 機能指数 1^n を受け取る．初めに，Setup_{sel} を実行し，鍵セット $(\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel})$ を得る．$\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}$ は，m 個の有限体 F_q 上の要素で構成されたベクトルである．次に，$\mathsf{Setup}_{\mathsf{PKE}}$ を実行し，鍵セット

$(\mathrm{PK}, \mathrm{SK})$ を得る．最後に，マスタ公開鍵 $\mathrm{MPK} = (\mathrm{PK})$ とマスタ秘密鍵 $\mathrm{MSK} = (\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}, \mathrm{SK})$ を出力する．

$\mathsf{Enc}(\mathrm{MPK}, m \in M_n) \to \mathrm{CT}$. Enc アルゴリズムは，マスタ公開鍵 MSK とメッセージ空間 M_n 上の平文 m を受け取る．初めに，選択的安全な関数型暗号の暗号化アルゴリズム Enc_{sel} に平文 m を入力し，算術回路 $\mathsf{Enc}_{sel}(x, m : r)$ で設計する．次に，DARE compiler に算術回路 $\mathsf{Enc}_{sel}(x, m : r)$ とセキュリティパラメータ 1^λ，算術回路の各ワイヤの取りうる上限値 U を入力し，エンコーダ Encd とデコーダ Decd を得る．この時，Encd は，線形行列の集合 W と m 個のアフィン関数 $L_i(x)$ で構成されている．その後，$\mathsf{Enc}_{\mathsf{PKE}}$ を実行し，アフィン鍵の集合を公開鍵 PK で暗号化し，c を得る．最後に，暗号文 $\mathrm{CT} = \{\mathsf{Decd}, W, \mathsf{c}\}$ を出力する．

$\mathsf{KeyGen}(\mathrm{MSK}, f \in F_n) \to \mathrm{SK}_f$. KeyGen アルゴリズムは，マスタ秘密鍵 MSK と関数空間 F_n 上の関数 f を受け取る．初めに，KeyGen_{sel} アルゴリズムを実行し，$\mathrm{SK}_{f,sel}$ を得る．次に，mq 個の PKE の秘密鍵から MPK_{sel} の要素の値に対応する m 個の秘密鍵を選択する．最後に，秘密鍵 $\mathrm{SK} = (\mathrm{MPK}_{sel}, \mathrm{SK}_{f,sel}, \mathrm{SK})$ を出力する．

$\mathsf{Dec}(\mathrm{SK}_{f,sel}, \mathrm{CT}) \to \tilde{m}$. Dec アルゴリズムは，秘密鍵 SK_f と暗号文 CT を受け取る．初めに，$\mathsf{Dec}_{\mathsf{PKE}}$ を実行し，c を復号し，m 個のアフィン関数 $L_i(x_i)$ を得る．次に，Encd に MPK_{sel} を入力し，m 個のアフィン鍵 L_i を得る．次に，Decd に W と L_i を入力し，暗号文 $\tilde{\mathrm{CT}}$ を得る．最後に，Dec_{sel} を実行し，暗号文 $\tilde{\mathrm{CT}}$ を復号した平文 $\tilde{m}$ を出力する．

3.2.1 一般的構成

Algorithm 5 $\mathsf{Setup}(1^\lambda, 1^n)$

$(\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}) \leftarrow \mathsf{Setup}_{sel}(1^\lambda, 1^n)$
$(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Setup}_{\mathsf{PKE}}(1^\lambda)$
$\mathrm{MPK} = \mathrm{PK}$
$\mathrm{MSK} = \{\mathrm{MPK}_{sel}, \mathrm{MSK}_{sel}, \mathrm{SK}\}$
return $(\mathrm{MPK}, \mathrm{MSK})$

Algorithm 6 $\mathsf{Enc}(\mathrm{MPK}, m \in M_n)$

$ckt = \mathsf{Enc}_{sel}(x, m : r)$
$(\mathsf{Encd}, \mathsf{Decd}) \leftarrow \mathsf{DARE\ compiler}(ckt, 1^\lambda, U)$
$\mathsf{c} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(\mathrm{PK}, \{L_1(x), ..., L_e(x)\})$
$\mathrm{CT} = \{\mathsf{Decd}, W, \mathsf{c}\}$
return CT

Algorithm 7 $\mathsf{KeyGen}(\mathrm{MSK}, f \in F_n)$

$\mathrm{SK}_{sel,f} \leftarrow \mathsf{KeyGen}_{sel}(\mathrm{MSK}_{sel}, f)$
$\mathrm{SK}_f = \{\mathrm{MPK}_{sel}, \mathrm{SK}_{sel,f}, \mathrm{SK}\}$
return SK_f

Algorithm 8 $\mathsf{Dec}(\mathrm{SK}_f, \mathrm{CT})$

$\{L_1(x), ...L_m(x)\} \leftarrow \mathsf{Dec}_{\mathsf{PKE}}(\mathrm{SK}, \mathsf{c})$
$\{L_1, ..., L_m\} \leftarrow \mathsf{Encd}(\mathrm{MPK}_{sel})$
$\tilde{\mathrm{CT}} \leftarrow \mathsf{Decd}(W, \{L_i, ...L_m\})$
$\tilde{m} \leftarrow \mathsf{Dec}_{sel}(\mathrm{SK}_{f,sel}, \tilde{\mathrm{CT}})$
return $y \in \{0, 1, \perp\}$

3.2.2 安全性証明の道筋と解決すべき点

本節では，検討案 2 の変換により得られる暗号スキームが準適応的安全性を満たすことの証明の道筋と解決すべき点を示す．

Theorem1 が成立することを証明するための道筋として，3 つの補題を証明することが必要であると考える．以下に補題と現在検討中の証明の方法を示す．

補題 1：FE に対する任意の多項式時間の準適応的 CPA 攻撃者 $\mathcal{A}$ を内部に雇うことで，PKE に対する多項式時間の Semantically Secure 攻撃者 $\mathcal{B}$ を構築できることを示す．

検討中の証明：初めに，$\mathcal{B}$ はチャレンジ公開鍵 PK^* を受け取る．次に，$\mathcal{B}$ は Setup_{sel} を走らせる．$\mathcal{A}$ へ MPK を送る．$\mathcal{A}$ はチャレンジメッセージ (m_0^*, m_1^*) を $\mathcal{B}$ へ送る．$\mathcal{B}$ は算術回路 $\mathsf{C}_{m_0^*}, \mathsf{C}_{m_1^*}$ を作成し，DARE compiler により $(\mathsf{Encd}_{m_0^*}, \mathsf{Encd}_{m_0^*})$ を計算する．$\mathcal{B}$ は自身のチャレンジメッセージ $(L^0 = \{L_0^0(x), \cdots, L_e^0(x)\}, L^1 = \{L_0^1(x), \cdots, L_e^1(x)\})$ を，チャレンジャへ送る．チャレンジャは，$b^* \xleftarrow{R} 0, 1$ を選択し，$\mathsf{c}_{b^*} \leftarrow \mathsf{Enc}_{\mathsf{PKE}}(L^{b^*}, PK^*)$ を計算し，$\mathcal{B}$ へ送る．$\mathcal{B}$ は，$\hat{b} \xleftarrow{R} 0, 1$ に選択し，暗号文 $\mathrm{CT}^* = (\mathsf{Decd}_{\hat{b}}, W_{\hat{b}}, ct_{b^*})$ を $\mathcal{A}$ へ送る．この時，$\hat{b} \neq b^*$ の場合，$\mathcal{A}$ は $\perp$ を出力する．$\hat{b} = b^*$ の場合，$\mathcal{A}$ は b' を出力する．$b' = \hat{b}$ の場合，$b^* = 0$ と推察する．それ以外の場合，$b^* = 1$ と推察する．

補題 2：FE に対する任意の多項式時間の準適応的 CPA 攻撃者 $\mathcal{A}$ を内部に雇うことで，DARE compiler に対する多項式時間の Privacy 攻撃者 $\mathcal{B}$ を構築できることを示す．

検討中の証明: 初めに，$\mathcal{B}$ は $\mathsf{Setup}_{\mathsf{PKE}}$ と Setup_{sel} を走らせる．$\mathcal{A}$ へ MPK を送る．$\mathcal{A}$ はチャレンジメッセージ (m_0^*, m_1^*) を $\mathcal{B}$ へ送る．$\mathcal{B}$ は，$b^* \xleftarrow{R} 0, 1$ 行い，算術回路 $\mathsf{C} = \mathsf{Enc}_{sel}(\cdot, m_b^*; r)$ を計算する．$\mathcal{B}$ は $\mathsf{C}, \mathrm{MPK}_{sel}$ を，チャレンジャへ送る．その後，$\mathcal{B}$ は Garbled Circuits$W$ と m 個のアフィン関数 $L_i(x)$ を受け取る．$\mathcal{B}$ は，アフィン関数を PKE で暗号化し，チャレンジ暗号文を作成し，$\mathcal{A}$ へ送る．$\mathcal{A}$ は b' を出力し，$\mathrm{Win}(b' = b)$ の場合，$\mathcal{B}$ は Encd により回路が Garbled されたと推測し，それ以外の場合，Sim によりシミュレートされたと推測する．

補題 3：FE に対する任意の多項式時間の準適応的 CPA 攻撃者 $\mathcal{A}$ を内部に雇うことで，FE_{sel} に対する多項式時間の選択的 CPA 攻撃者 $\mathcal{B}$ を構築できることを示す．

検討中の証明: 初めに，$\mathcal{B}$ は $\mathsf{Setup}_{\mathsf{PKE}}$ を走らせる．$\mathcal{A}$ へ MPK を送る．$\mathcal{A}$ はチャレンジメッセージ (m_0^*, m_1^*) を $\mathcal{B}$ へ送る．$\mathcal{B}$ は，自身のチャレンジメッセージとして (m_0^*, m_1^*)

をチャレンジャへ送る．チャレンジャは $b^* \xleftarrow{R} 0,1$ を選択し，Setup_{sel} を走らせ，$\mathsf{c}^* \leftarrow \mathsf{Enc}(\mathrm{MPK}_sel, m_b^*)$ 計算する．チャレンジャは，$\mathsf{c}^*, \mathrm{MPK}_{sel}$ を $\mathcal{B}$ へ送る．$\mathcal{B}$ は，$\hat{b} \xleftarrow{R} 0,1$ を選択し，DARE compiler で $\mathsf{Sim}_{\hat{b}}$ を生成する．チャレンジャは，$\mathsf{c}^* \leftarrow \mathsf{Enc}_{sel}(\mathrm{MPK}_{sel}, m_b^*)$ を計算し，$\mathcal{B}$ へ送る．$\mathcal{B}$ は，$(W, L_i(x)) \leftarrow \mathsf{Sim}_{\hat{b}}(\mathsf{c}^*; r)$ を生成する．この時，$\hat{b} = b^*$ ならば，$\mathcal{A}$ は b' を出力する．それ以外ならば，$\mathcal{A}$ は $\perp$ を出力する．$\mathcal{B}$ は，受け取った b' をそのまま自身の推測として出力する．

上記の手法を証明することができれば，検討案２の変換後に得られる暗号スキームが準適応的安全であることを証明できると考える．

ただし，解決すべき点として，DARE compiler により得られる Sim は，関数の値を出力する．我々の手法の補題 2,3 において Sim は関数を出力する必要がある．そのため関数の値から関数への変換を考えなけらえばならない．ただし，変換後の関数は，元の関数を完全に再現する必要はなく，入力 MPK_{sel} に対して同じ出力を返す関数で十分であると考える．

4. 性能評価

本節では，オリジナルの GKW 変換により得られる準適応的安全な FE の性能と我々の検討案により得られる準適応的安全な FE の性能の評価比較を示す．

オリジナルの GKW 変換は，公開鍵 MPK は MPK_{sel} のビット長 $l(n)$ の定数倍と等しい個数の PKE 公開鍵 PK の集合であるため，公開鍵長は機能指数 n の多項式となった．同様に，秘密鍵長，暗号鍵長はビット長 $l(n)$ を参照するため，機能指数 n の多項式となった．

検討案１は，F_q 状の全ての元 $0, \cdots, q-1$ の入力に対応するワイヤ鍵を生成するため，公開鍵長と暗号文長はセキュリティパラメータ λ の指数関数となった．秘密鍵長は，MPK_{sel} の要素数を参照するため，機能指数 n の多項式となった．

検討案２は，公開鍵長，秘密鍵長，暗号文長全てにおいて定数倍となった．

得られた結果から，検討案１は，公開鍵長，暗号文長指数的に爆発してしまうため，今後の研究に繋げることが困難であることがわかった．検討案２は，公開鍵長，秘密鍵長，暗号文長全てにおいてオリジナルの GKW 変換からの短縮が見込めることがわかった..

表 1 変換後の準適応的安全な FE の性能評価

	公開鍵長	秘密鍵長	暗号文長
論理回路 [3]	$O(n)$	$O(n)$	$O(n)$
検討案１	$O(n2^\lambda)$	$O(n)$	$O(n2^\lambda)$
検討案２	$O(1)$	$O(1)$	$O(1)$

5. まとめと今後の課題

我々は算術回路に対する GKW 変換について２つの検討案を示した． F_q の元に対応したワイヤ鍵を生成する検討案については，一般的な構成法を示し，安全性証明を行った．しかしながら，公開鍵長，暗号文長が指数的に爆発してしまうことがわかった．ワイヤ鍵を生成する関数 f に重点を置く検討案については，一般的な構成法を示し，安全性証明の筋道と解決すべき点を示した．こちらはオリジナルの GKW 変換と比較して，優位な結果がしめされた．今後の課題は，検討案２の安全性証明を完了することである．具体的な課題としては，DARE compiler の Sim が出力する関数の値から，特定の入力に対する出力が同じ結果となる関数に変換する方法を模索する．また，アルゴリズムの計算効率についても性能評価を行う．

謝辞

本研究に関わる CSS2020 での発表に対しコメント下さった草川恵太様，北川冬航様に深謝します．

参考文献

[1] B. Applebaum, Y. Ishai, and E. Kushilevitz.
How to garble arithmetic circuits.
SIAM J. Comput., 43(2):905–929, 2014.

[2] J. Chen and H. Wee.
Semi-adaptive attribute-based encryption and improved delegation for boolean formula.
In M. Abdalla and R. D. Prisco, editors, *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, volume 8642 of *Lecture Notes in Computer Science*, pages 277–297. Springer, 2014.

[3] R. Goyal, V. Koppula, and B. Waters.
Semi-adaptive security and bundling functionalities made generic and easy.
In M. Hirt and A. D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 361–388, 2016.

[4] R. Ostrovsky, A. Sahai, and B. Waters.
Attribute-based encryption with non-monotonic access structures.
In *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, pages 195–203, 2007.

[5] O. Regev.
On lattices, learning with errors, random linear codes, and cryptography.
J. ACM, 56(6):34:1–34:40, 2009.