ハードウエアセキュリティと暗号ハードウエア設計

池田　誠
東京大学大学院工学系研究科
附属システムデザイン研究センター(d.lab)

謝辞

本講義で紹介する研究成果の一部は、国立研究開発法人新エネルギー・産業技術総合開発機構(NEDO)の複数のプロジェクトによる支援によるものです。

また、紹介するチップの過半はVDECのチップ試作プログラムを通して行ったものであり、Cadence、Synopsys、メンターグラフィックス（現在はSiemens）によるアカデミックプログラム＆拡張アカデミックプログラムの枠組みにより提供されたEDAツールを用いて設計したものです。

また、東京大学電気系工学専攻池田研究室の現在および卒業生により得られた成果となっています。

内容

- 背景
  - セキュリティとハードウエア
  - 共通鍵暗号（ブロック暗号）
  - 公開鍵暗号と鍵管理＆ディジタル署名
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

ハードウエアセキュリティとは

- 暗号アルゴリズム
  - 情報の秘匿化(暗号化)
  - 情報の正当性の証明(署名)
  - 鍵情報管理
- 暗号アルゴリズムの安全性
  - 解読?
  - 攻撃耐性
- ハードウエアトロイ
- 乱数
- 計測セキュリティ

暗号エンジンのハードウエア実装

- 新規暗号の考案・・数学――＞ソフト実装
- 暗号エンジンのハードウエア実装・・
  - アルゴリズムの理解（数学）
  - アーキテクチャの検討（情報＝数学）
  - 物理的制約（遅延時間、ハードウエア量、消費電力等）を考慮した最適化
    - 特に微細CMOSで単純にスケールした性能見積もりは有効ではない
  - 実現の可能性を示すことに大きな意義がある

暗号エンジンの実装

- ソフトウエア
  - いつでも書き換え可能：アルゴリズム、パラメータの変更に柔軟に対応可能
  - 遅い
- FPGA
  - 書き換え可能
  - そこそこ速い
- 専用ハードウエア（ASIC）
  - 後から書き換えできない
  - 速い

暗号エンジンの実装

- ソフトウエア
  - いつでも書き換え可応可能
  - ~~遅い~~
- FPGA
  - 書き換え可能
  - そこそこ速い
- 専用ハードウエア(
  - 後から書き換えでき
  - ~~速い~~

✓NREは小さいが、運用のための電気代などが高額になる

✓スループットは並列計算（GPGPUを並べる、スパコンを活用）により向上可能
✓Latencyを稼ぐには効率的なアルゴリズム構築が不可欠
✓最先端プロセスを利用できるので高速化が可能

✓最先端プロセスを利用できるので高速化が可能（株取引などで活用が進んでいる）
✓並列化も可能、リピートのコストは高い

✓規格化が進めばパラメータ可変にする程度でも問題ない
✓汎用エンジンとして設計することで多種に対応させる

✓最先端プロセスはNREが高額で利用できない
✓見込みを示すことが重要？

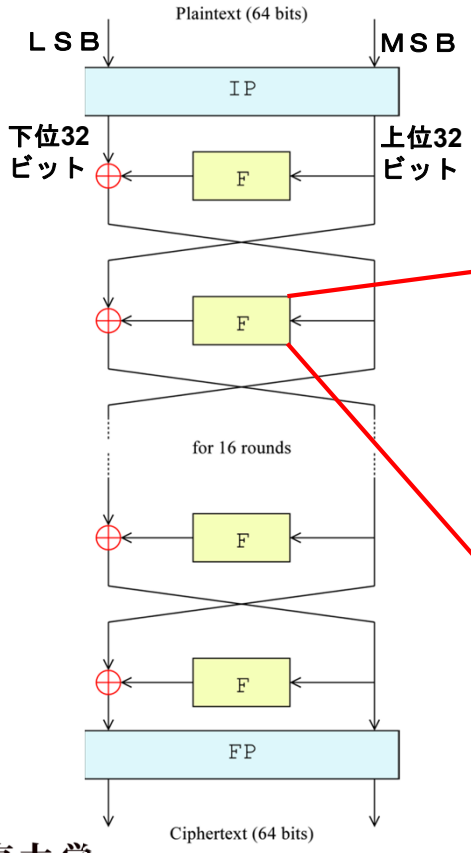
ところで暗号ハードウエア実装・・

- 回路系の学会においてはほとんどはAESなど共通鍵方式の高速化・小面積化を狙ったもの
  - 暗号化向け
- ディジタル署名・鍵交換などを考えると公開鍵方式の高速化・小面積化が不可欠
- 暗号・署名の更なる広範囲での利用に向けては高機能暗号の実用化・高速処理が不可欠

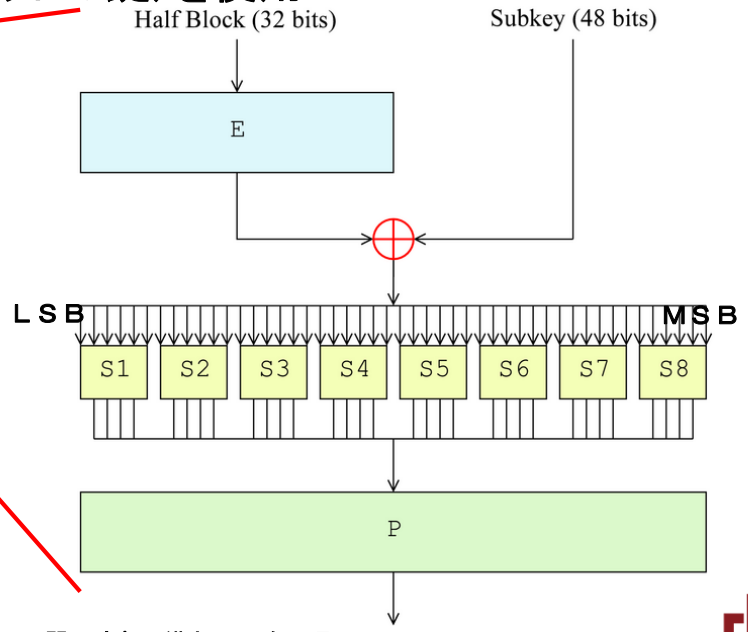
内容

- 背景
  - セキュリティとハードウエア
  - 共通鍵暗号（ブロック暗号）
  - 公開鍵暗号と鍵管理＆ディジタル署名
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

暗号化とDES他



- 共通鍵暗号、ブロック暗号の一種
- 1976年に米国で採用
- 64ビットを1ブロックとして、64ビットの鍵（実際には56ビットの鍵）を使用



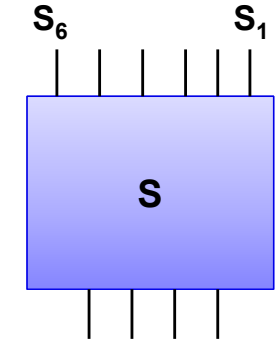
IP：初期転置: 64bit → 64bit & Sボックス処理：48bit → 32bit

入力X
64bit

出力Y
64bit

x1	x2	x3	x4	・	・	・	・	・	x61	x62	x63	x64

y1	y2	y3	y4	y5	y6	y7	y8
x58	**x50**	**x42**	**x34**	**x26**	**x18**	**x10**	**x2**
y9	y10	y11	y12	y13	y14	y15	y16
x60	**x52**	**x44**	**x36**	**x28**	**x20**	**x12**	**x4**
y17	y18	y19	y20	y21	y22	y23	y24
x62	**x54**	**x46**	**x38**	**x30**	**x22**	**x14**	**x6**
y25	y26	y27	y28	y29	y30	y31	y32
x64	**x56**	**x48**	**x40**	**x32**	**x24**	**x16**	**x8**
y33	y34	y35	y36	y37	y38	y39	y40
x57	**x49**	**x41**	**x33**	**x25**	**x17**	**x9**	**x1**
y41	y42	y43	y44	y45	y46	y47	y48
x59	**x51**	**x43**	**x35**	**x27**	**x19**	**x11**	**x3**
y49	y50	y51	y52	y53	t54	y55	y56
x61	**x53**	**x45**	**x37**	**x29**	**x21**	**x13**	**x5**
y57	y58	y59	y60	y61	y62	y63	y64
x63	**x55**	**x47**	**x39**	**x31**	**x23**	**x15**	**x7**



S_6S_1

{ 14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7,
0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8,
4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0,
15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13, },

{ 15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10,
3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5,
0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2, 15,
13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9, },

$S_5S_4S_3S_2$

{ 10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8,
13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1,
13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7,
1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12, },

{ 7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,
13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,
10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,
3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14, },

{ 2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9,
14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6,
4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14,
11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3, },

{ 12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11,
10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8,
9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6,
4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13, },

{ 4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1,
13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6,
1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2,
6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12, },

{ 13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7,
1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2,
7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8,
2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11, },

内容

- 背景
  - セキュリティとハードウエア
  - 共通鍵暗号（ブロック暗号）
  - 公開鍵暗号と鍵管理＆ディジタル署名
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

公開鍵暗号・RSAと高速化（モンゴメリ乗算）

- p=17(5bit), q=19(5bit) → n=323 (9bit)
- e=2^2+1=5
- L= ϕ(p−1)(q−1) =144 (p−1=16, q−1=18の最小公倍数)
- d*e + L*v=1を満たすd: (d=29, v=1)
- p, q, d: 秘密鍵、n, e: 公開鍵
  - $c = m^e \bmod n$ を暗号文とする
  - $m=c^d \bmod n$ により復号

    べき乗剰余演算

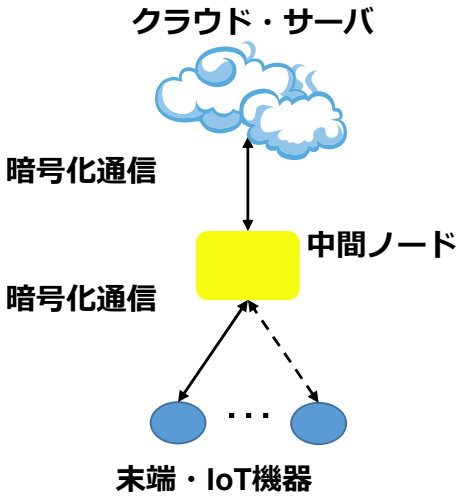
- モンゴメリ表現：整数a: $0 \leqq a < M$に対して、ある定数Rをかけた表現$A=aR \bmod N$をモンゴメリ表現と呼ぶ
  - $c=a \pm b$ → $C=A \pm B$, $c=CR^{-1} \bmod N$: $C \geqq N$, $C<0$の時にNを加減算すればよい
  - $c=a*b$ → $C=A*B/R$ → 以下のモンゴメリリダクションMR()を考える
- $T \bmod M$に対して
  - $0 \leqq T \leqq NR$に対して
  - MR (T) = $TR^{-1} \bmod N$を考える
  - $RR^{-1}-N'N=1$(N'N=−1 mod R, $RR^{-1}=1 \bmod N$) なるN', R^{-1}を求める：たとえば$R=2^4$のとき$N'=5$, $R^{-1}=101$ (N=323)
  - MR(T)=(T+($TN' \bmod R$)N)/R
  - aをモンゴメリ表現Aにするには $A=aR \bmod N=(aR^2)R^{-1} \bmod N$ → A=MR(a($R^2 \bmod N$)), つまりあらかじめ$R_2=R^2 \bmod N$を計算しておきA=MR(aR_2)求めればよい
  - 逆変換(A→a)は $a=aRR^{-1} \bmod N$=MR(A)
  - $c = m^e \bmod n$ → M_M=MR(m)に対してM_M^e を求め、c = MR (M_M^e)とする

楕円曲線暗号

- RSA:現時点では2048bit以上の鍵使用が求められている一＞演算量が非常に大きくなっている
- 鍵長の短い暗号として、楕円曲線暗号が知られている

RSA [bit]	楕円曲線 [bit]
1024	160-223
2048	224-255
3072	256-383

低電力・小面積：IoT機器向け鍵管理



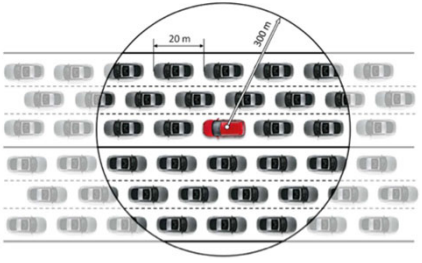
- IoT機器・中間ノード・クラウド/サーバ間の通信の安全性の担保
  - データの秘匿性保証のための暗号化
  - データの真正性保証のための署名付与
  - 安全な鍵管理

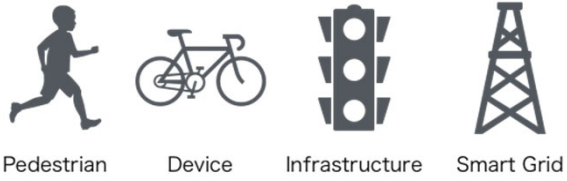
→ 楕円曲線に基づく暗号の高速・低遅延・低電力・底面積での実装が不可欠

高速性：車載通信（V2X）向けディジタル署名

- V2Xの安全性への要求
  - (V2V) ～１秒間に1,000回程度の署名生成・検証
  - (V2X) ～１秒間に10,000回程度の署名生成・検証
  - 低遅延・高スループットでの実現が不可欠
  - 軽量暗号？

→楕円曲線に基づく暗号の高速・低遅延実装が不可欠

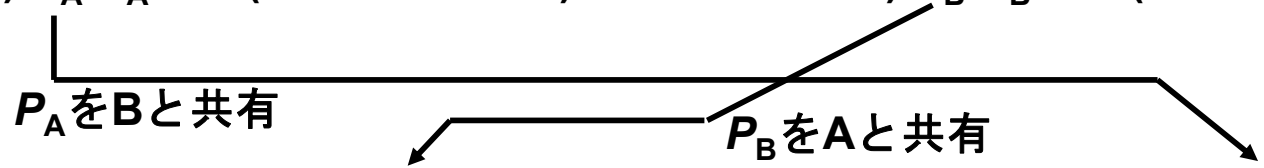




ECDHによる鍵交換

A:末端機器
1) 乱数 r_A 生成
2) $P_A=r_A$ x G (スカラ倍算1回)

B:サーバ
1) 乱数 r_B 生成
2) $P_B=r_B$ x G (スカラ倍算1回)



3) $Q\,(x_Q, y_Q) = r_A$ x P_B (スカラ倍算1回)

3) $Q\,(x_Q, y_Q) = r_B$ x P_A (スカラ倍算1回)

ここでA, Bいずれの$Q\,(x_Q, y_Q) = r_A$ x P_B
$= r_B$ x $P_A = r_A$ x r_B x G

4) 共通秘密鍵 Key = SHA256(x_Q)

4) 共通秘密鍵 Key = SHA256(x_Q)

ECDSA(楕円曲線暗号署名):生成

- 楕円曲線に対して、基点Gおよびn*G=Oを満たす位数nの巨大な素数を決めておく
- [1,n−1]の秘密鍵d_A, 公開鍵 $Q_A=d_A*G$を生成 (*スカラ倍算)
- メッセージmに対して e=HASH(m)を計算
  HASHはSHAなどのハッシュ関数
- eのMSB側のLnビット(位数nとなるビット数)をzとする
- [1,n−1]の任意の整数kを選択
- 曲線上の点(x1, y1)=k*Gを計算
- r=x1 mod nを求める (ただしr=0となる場合にはkを選択しなおす
- $s=k^{-1}(z+rd_A) \bmod n$を計算
- (r,s)をmに対する署名とする

ECDSA(楕円曲線暗号署名):検証

- r, sが[1, n−1]にあることを確認:ない場合不正な署名
- e=HASH(m)を求める
- eのMSB側のLnビット(位数nとなるビット数)をzとする
- $w=s^{-1} \bmod n$を求める
- $u1=zw \bmod n$, $u2=rw \bmod n$を求める
- 曲線上の点$(x1, y1)=u1*G + u2* Q_A$を計算
- $r=x1 \bmod n$であれば正当な署名

内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
  - 数学的解釈
  - 物理的解釈
  - アーキテクチャ＆スケジューリング
  - 実装・評価例
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

楕円曲線と離散対数問題

楕円曲線上で加群となるような点の**加算・2倍算**

↓

加算・2倍算の繰り返しによりベースポイントGを**スカラーk倍**

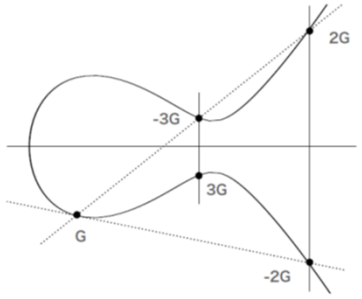
↓

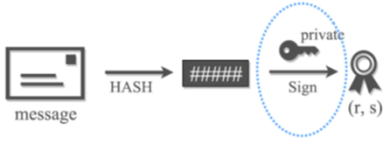
kGからkを求める楕円曲線上の離散対数問題（**ECDLP**）

↓

ECDLPによる安全な**署名生成**

幾何学的に演算した例（k = 3）





署名生成にスカラー倍算が用いられる

演算：剰余演算→ モンゴメリ乗算の利用の有無

点の演算：加算・定数乗算（倍算）

$$G_1(x_1, y_1) + G_2(x_2, y_2) \rightarrow G_3(x_3, y_3)$$

$$x_3 = \lambda_3^2 - x_2 - x_1$$

$$y_3 = \lambda_3(x_1 - x_3) - y_1$$

$$\lambda_3 = \frac{y_2 - y_1}{x_2 - x_1}$$

$$2G_1(x_1, y_1) \rightarrow G_4(x_4, y_4)$$

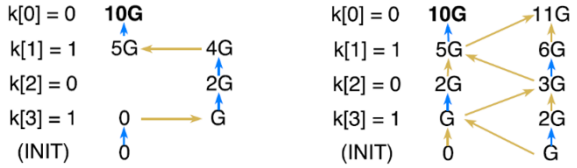
$$x_4 = \lambda_4^2 - 2x_1$$

$$y_4 = \lambda_4(x_1 - x_4) - y_1$$

$$\lambda_4 = \frac{3x_1^2 - a}{2y}$$

倍算の演算： $Q = kG = \sum_{i=0}^{n-1} k_{i-1} \cdot 2^{i-1} G$

LtR(RtL) / モンゴメリラダー



ただし、ここの「数」はすべてある素数pの剰余系の数値である

楕円曲線上の基本演算

楕円曲線上の任意の点Pに対してP+O=Pとなる点Oを零元O=(∞, ∞)

楕円曲線上の任意の点Pに対してkP (kは整数)をスカラー倍算とよび、点の加算と２倍算で実現できる

Input: k, P

Output: kP

1: $Q = O$
2: **for** $i = n - 1$ **to** 0 **do**
3: $\quad Q = 2Q$
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad Q = Q + P$
6: $\quad$ **endif**
7: **endfor**
8: **return** Q

バイナリ法

Input: k, P

Output: kP

1: $Q = O$
2: $R = P$
3: **for** $i = n - 1$ **to** 0 **do**
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad Q = Q + R$
6: $\quad\quad R = 2R$
7: $\quad$ **else**
8: $\quad\quad R = Q + R$
9: $\quad\quad Q = 2Q$
10: $\quad$ **endif**
11: **endfor**
12: **return** Q

モンゴメリラダー法

座標系の選択

- アフィン座標系：(x_i, y_i)
  - ✓除算が必要
- 射影座標系(Projective座標系)：$(x_i, y_i) = \left(\frac{X_i}{Z_i}, \frac{Y_i}{Z_i}\right)$
  - ✓加算に有利
- ヤコビアン座標系(Jacobian座標系)：
  - ✓２倍算に有利

$$(x_i, y_i) = \left(\frac{X_i}{Z_i^2}, \frac{Y_i}{Z_i^2}\right)$$

- 演算器の基数の選定
  - 256bitの演算を何bitの演算器で実行するか
    CPUの場合には 64bit固定、
    GPUの場合は？
    FPGAの場合48bitDSPを利用することが最適？
    ASICの場合：？？？

座標系	加算	2 倍算	加算+2 倍算
アフィン座標	$2M + S + I$	$2M + 2S + I$	$4M + 3S + 2I$
プロジェクティブ座標	$12M + 2S$	$7M + 5S$	$19M + 7S$
ヤコビアン座標	$11M + 5S$	$2M + 8S$	$13M + 13S$

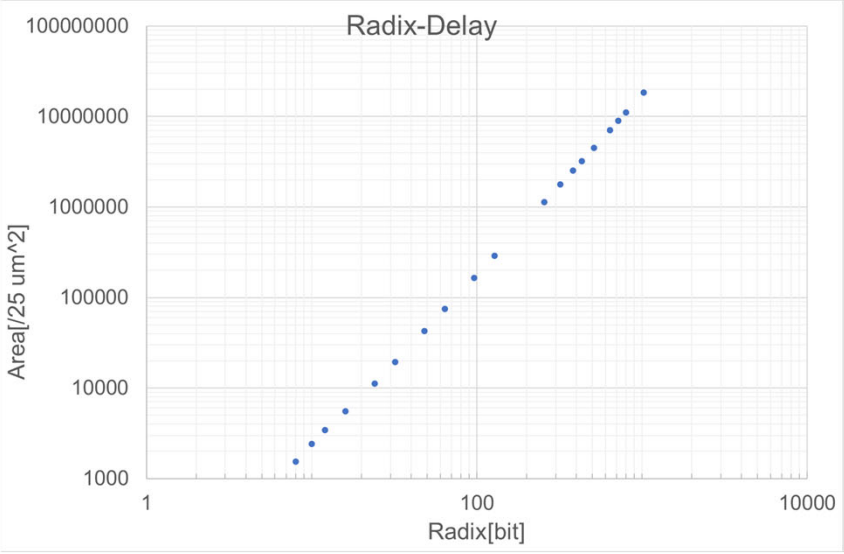
M: 乗算
S: 自乗算
I: 逆数演算

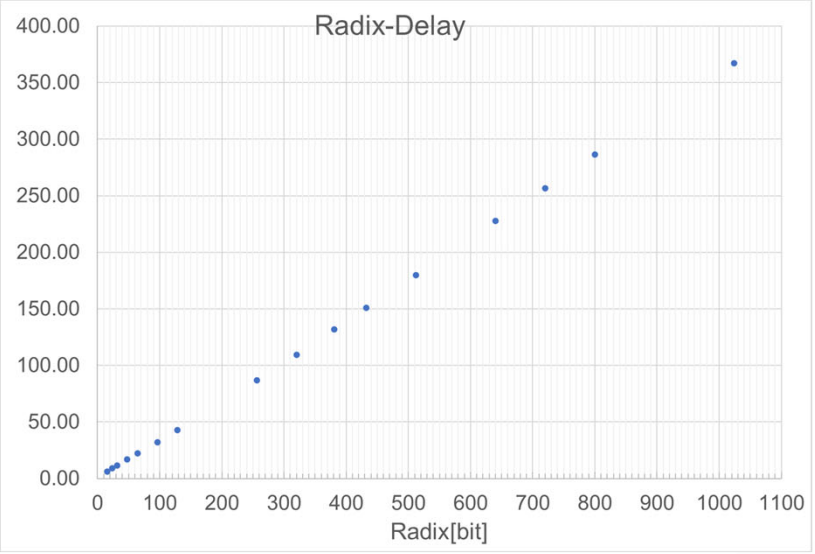
内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
  - 数学的解釈
  - 物理的解釈
  - アーキテクチャ＆スケジューリング
  - 実装・評価例
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

基数の選択:乗算器

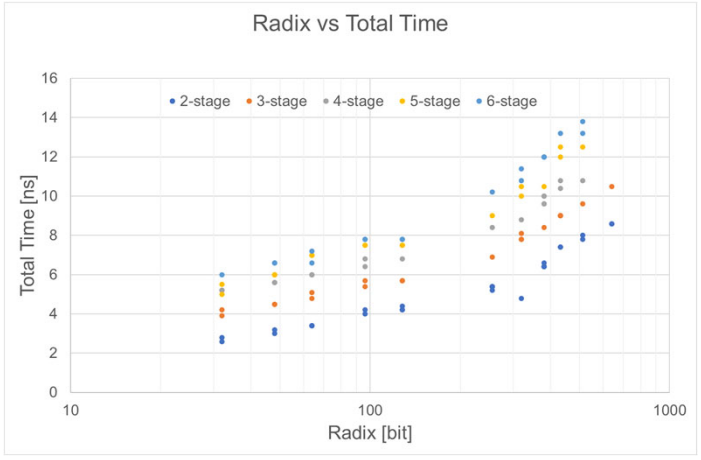
- 演算器の基数の選定
  - 256bitの演算を何bitの演算器で実行するか
    - ✓ CPU：64bit固定
    - ✓ GPU：？
    - ✓ FPGA：48bitDSPの利用が最適？
    - ✓ ASIC：？？？

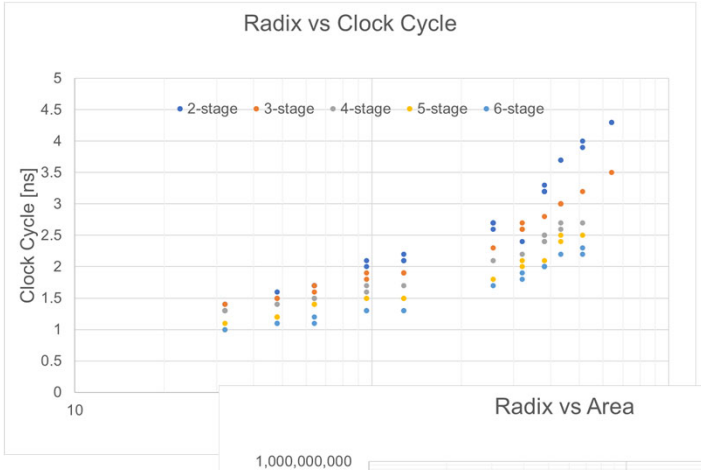


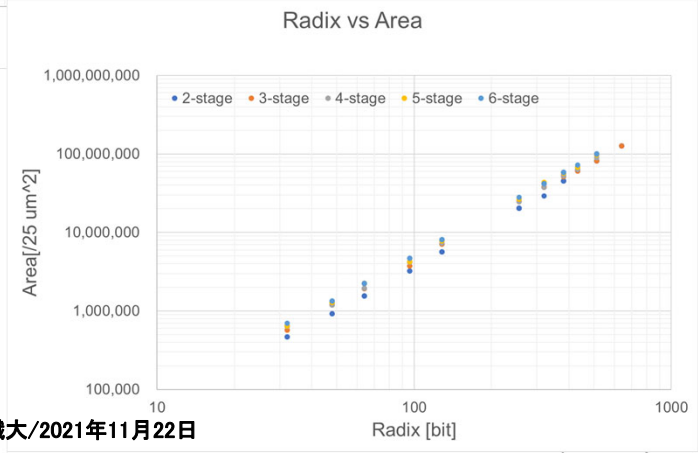


演算器(乗算器)とパイプライン段数

- **256bitの乗算器：何段パイプラインが最適？**







基数の選択：加算器のビット幅と遅延・面積

Architecture	Classification	Logic Levels	Max Fanout	Tracks	Cells
Carry-Ripple		$N-1$	1	1	N
Carry-Skip ($n=4$)		$N/4+5$	2	1	$1.25N$
Carry-Increment ($n=4$)		$N/4+2$	4	1	$2N$
Carry-Increment (variable group)		$\sqrt{2N}$	$\sqrt{2N}$	1	$2N$
Brent-Kung	$(L-1, 0, 0)$	$2\log_2 N-1$	2	1	$2N$
Sklansky	$(0, L-1, 0)$	$\log_2 N$	$N/2+1$	1	$0.5\,N\log_2 N$
Kogge-Stone	$(0, 0, L-1)$	$\log_2 N$	2	$N/2$	$N\log_2 N$
Han-Carlson	$(1, 0, L-2)$	$\log_2 N+1$	2	$N/4$	$0.5\,N\log_2 N$
Ladner Fischer ($l=1$)	$(1, L-2, 0)$	$\log_2 N+1$	$N/4+1$	1	$0.25\,N\log_2 N$
Knowles [2,1,…,1]	$(0, 1, L-2)$	$\log_2 N$	3	$N/4$	$N\log_2 N$

遅延：O(*n*) or O(log *n*)

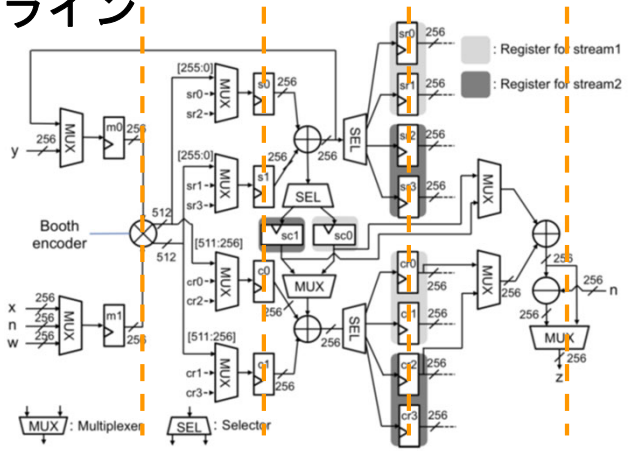
面積：O(*n*) or O(*n*log *n*)

内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
  - 数学的解釈
  - 物理的解釈
  - アーキテクチャ＆スケジューリング
  - 実装・評価例
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

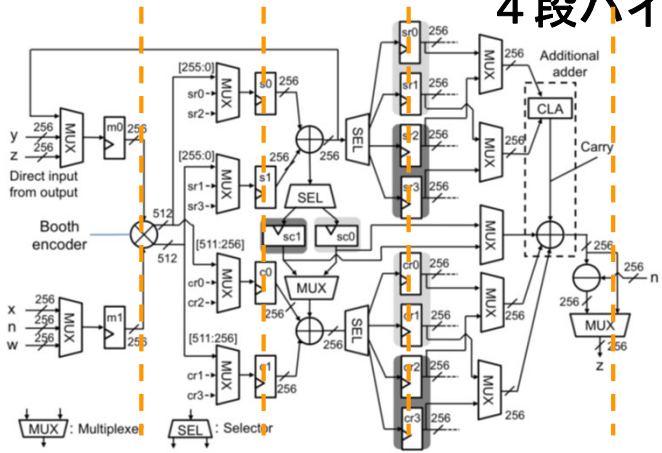
モンゴメリ乗算器の最適化

4段パイプライン



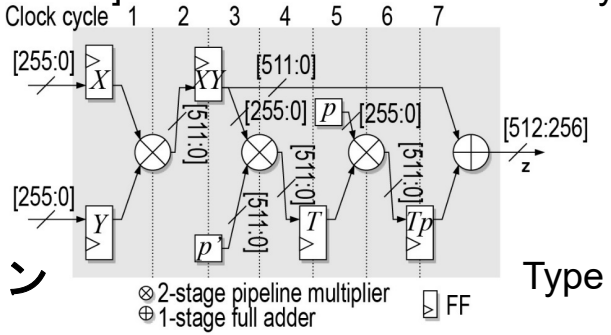
Type 1 [A-SSCC 2016]

4段パイプライン



Type 2 [IEICE 2016]

7段パイプライン

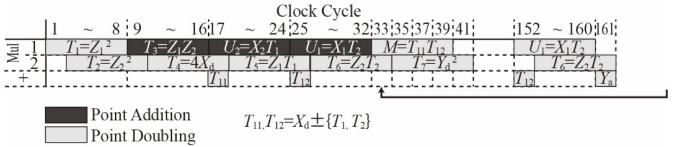


Type 3 [A-SSCC 2018]

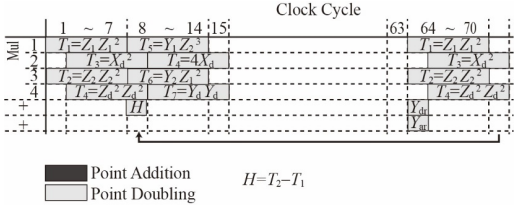
アーキテクチャ・スケジュール

Require:
$X, Y \in \mathbb{F}_p$,
$R = 2^{\lfloor \log_2 p \rfloor}$,
$p' = -p^{-1} \bmod R$
Ensure: $Z = XYR^{-1} \bmod p$
1: $T \Leftarrow (XY \bmod R) \cdot p' \bmod R$
2: $Z \Leftarrow (XY + Tp)/R$
3: **if** $Z \geq p$ **then**
4: $\quad Z \Leftarrow Z - p$
5: **end if**
6: **return** Z

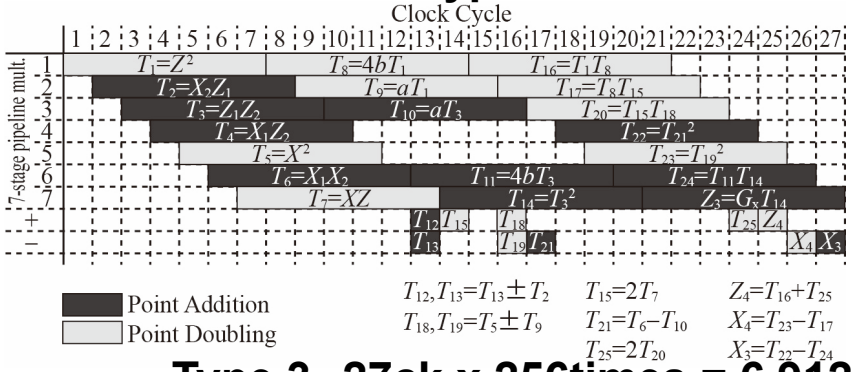
モンゴメリ乗算アルゴリズム



Type 1　128ck x 256times = 32,768ck



Type 2　56ck x 256times = 14,336ck



Type 3　27ck x 256times = 6,912ck

モンゴメリ乗算アルゴリズムと数式

Case 1

Assumptions: 4*a24=a+2

Cost: 6M + 4S + 1*a24 + 8add

$A = X2+Z2$
$AA = A^2$
$B = X2-Z2$
$BB = B^2$
$E = AA-BB$
$C = X3+Z3$
$D = X3-Z3$
$DA = D*A$
$CB = C*B$
$X5 = Z1*(DA + CB)^2$
$Z5 = X1*(DA - CB)^2$
$X4 = AA*BB$
$Z4 = E*(BB+a24*E)$

Case 2

Assumptions: 4*a24=a+2

Cost: 8M + 4S + 1*a24 + 7add

t0 = X3 - Z3
t1 = X2 + Z2
t2 = X3 + Z3
t3 = X2 - Z2
t4 = t2 * t3
t5 = t0 * t1
t6 = t5 + t4
t7 = t6 ^ 2
X5 = Z1 * t7
t8 = t5 - t4
t9 = t8 ^ 2
Z5 = X1 * t9
t10 = t1 ^ 2
t11 = t3 ^ 2
X4 = t10 * t11
t12 = X2 * Z2
t13 = 4 * t12
t14 = a24 * t13
t15 = t11 + t14
Z4 = t13 * t15

Case 3

Cost: 10M + 5S + 1*a24 + 5add

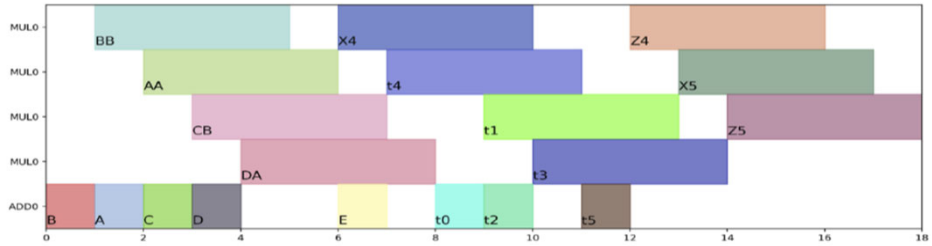
t0 = Z2 * Z3
t1 = X2 * X3
t2 = t1 - t0
t3 = t2 ^ 2
X5 = Z1 * t3
t4 = Z2 * X3
t5 = X2 * Z3
t6 = t5 - t4
t7 = t6 ^ 2
Z5 = X1 * t7
t8 = X2 ^ 2
t9 = Z2 ^ 2
t10 = t8 - t9
X4 = t10 ^ 2
t11 = X2 * Z2
t12 = a * t11
t13 = t8 + t12
t14 = t13 + t9
t15 = Z2 * t14
t16 = X2 * t15
Z4 = 4 * t16

Curve 25519

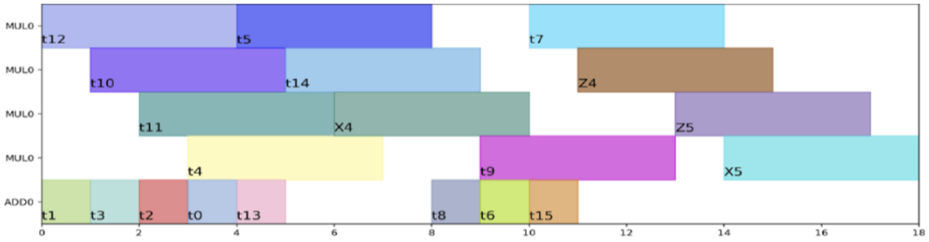
数式とスケジュール結果（モンゴメリ乗算：4段パイプライン）

Which one the best?

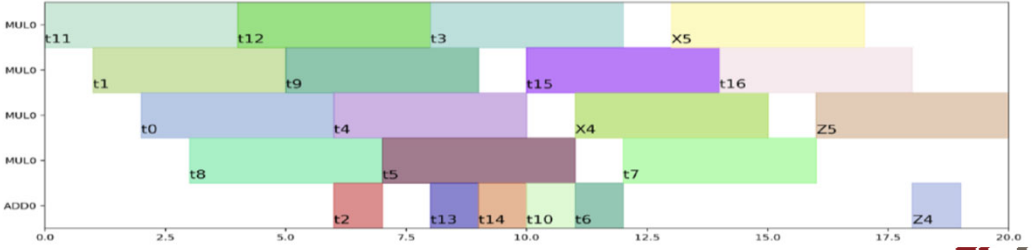
Case 1



Case 2



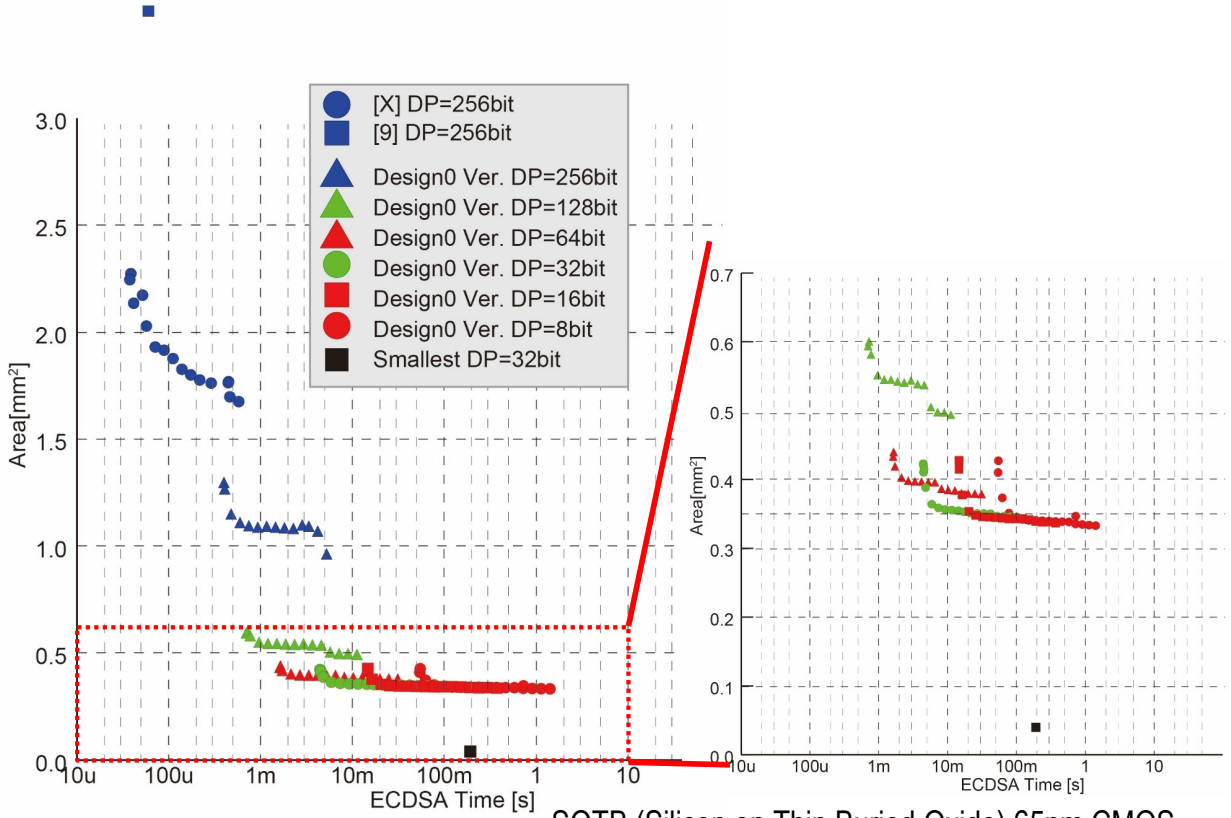
Case 3



内容

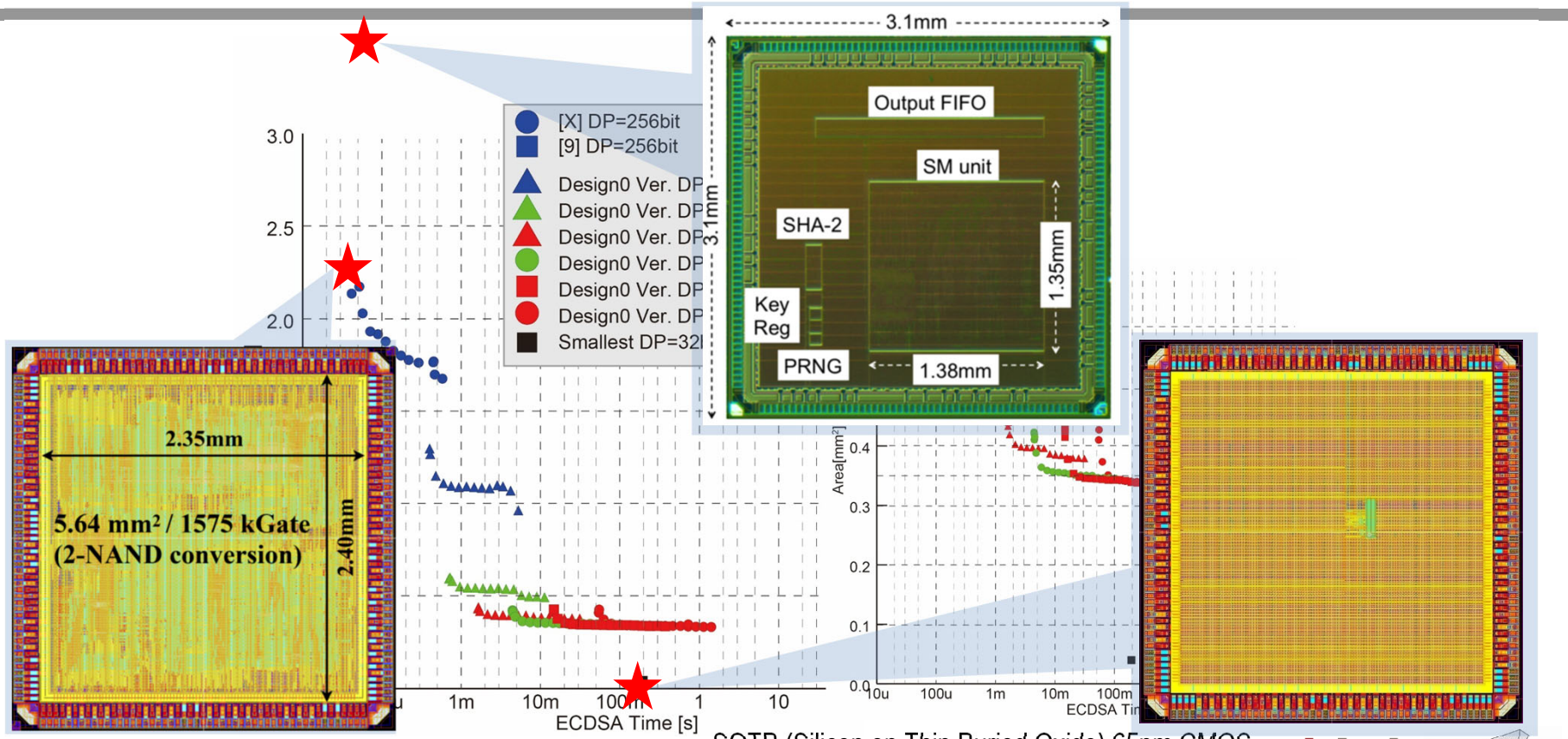
- 背景
- 楕円曲線暗号向けハードウエアの実現
  - 数学的解釈
  - 物理的解釈
  - アーキテクチャ＆スケジューリング
  - 実装・評価例
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

種々の設計例：遅延時間と面積



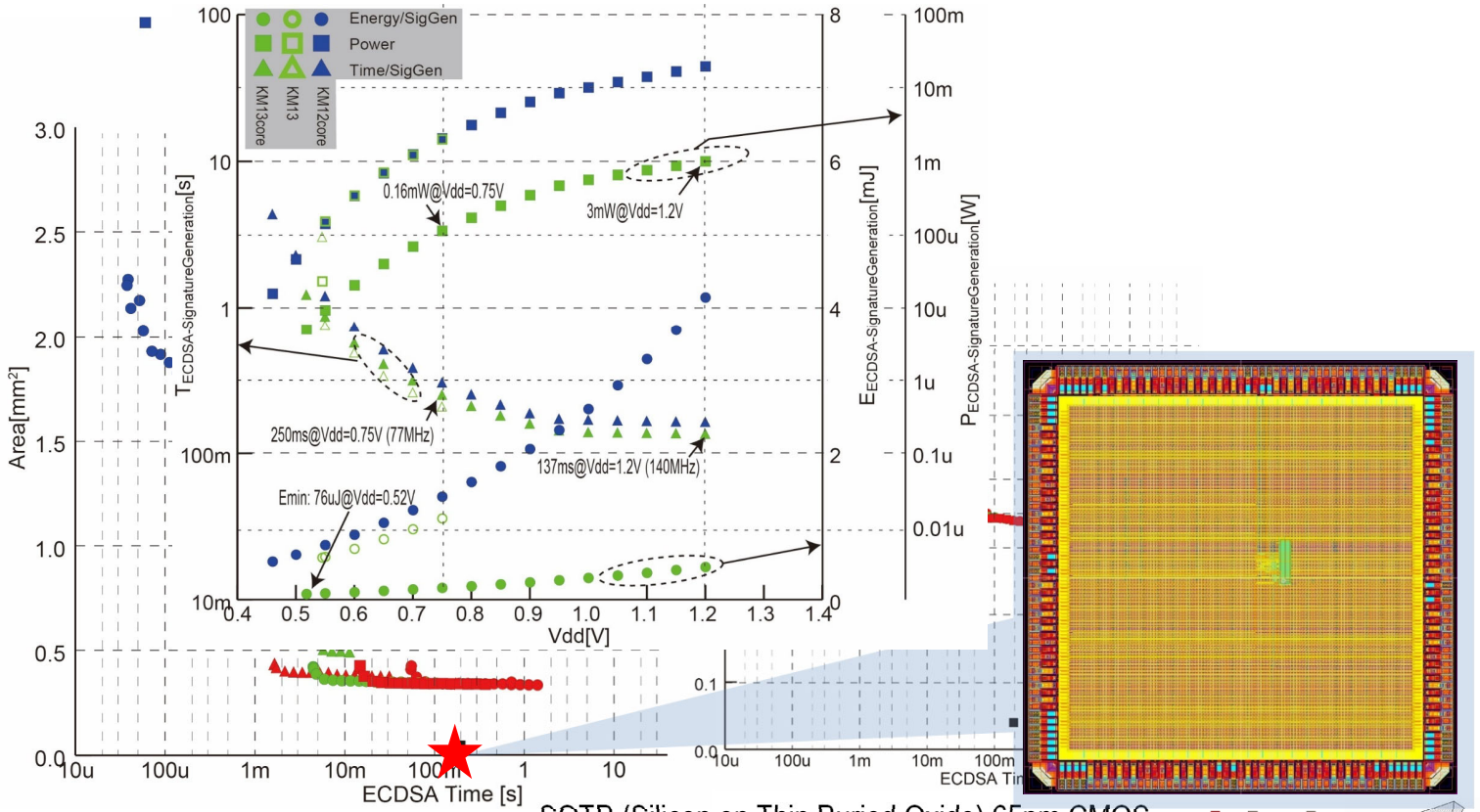
SOTB (Silicon on Thin Buried Oxide) 65nm CMOS

種々の設計例：実測例



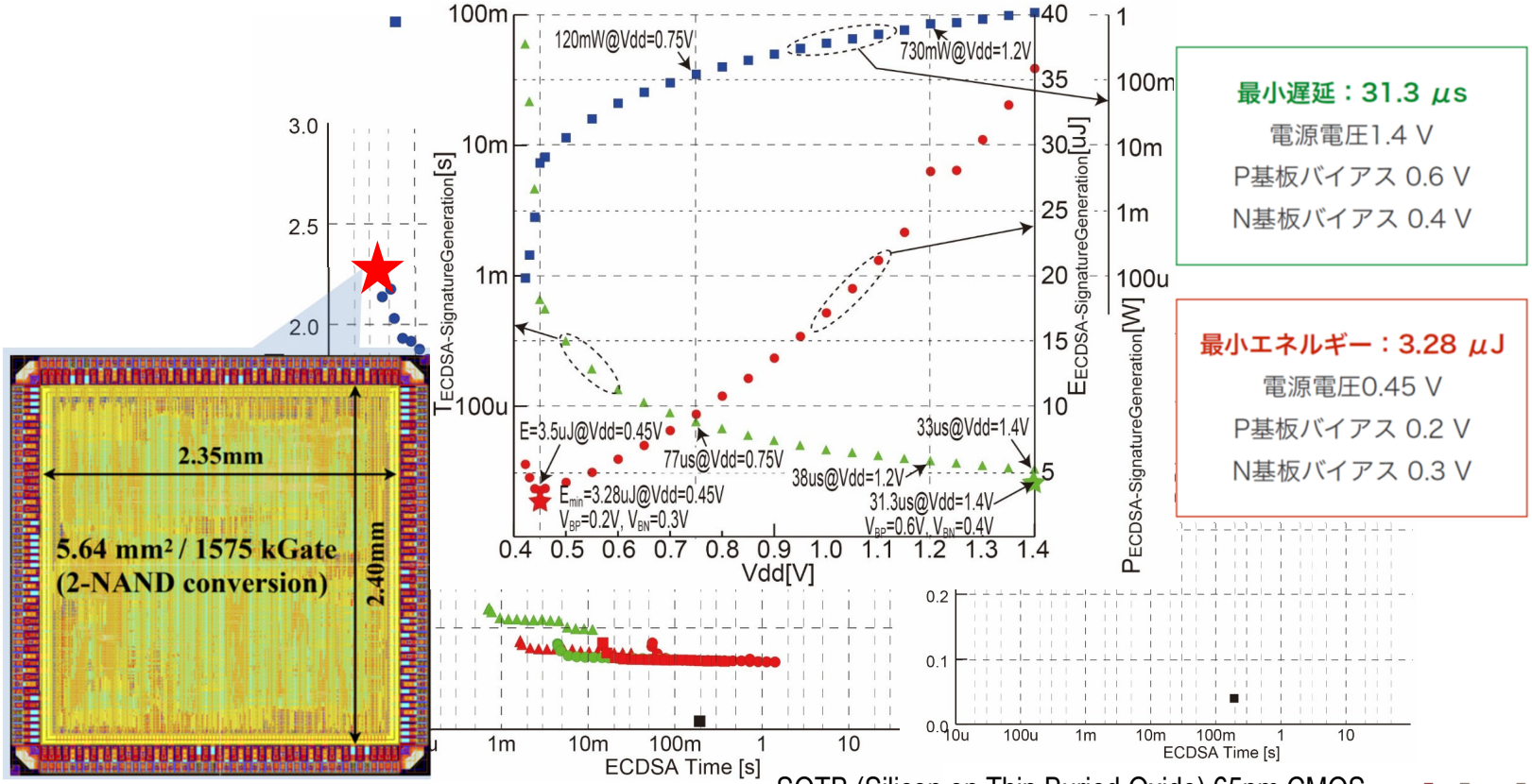
SOTB (Silicon on Thin Buried Oxide) 65nm CMOS

種々の設計例：実測例：最小面積版



SOTB (Silicon on Thin Buried Oxide) 65nm CMOS

種々の設計例：実測例：最速版

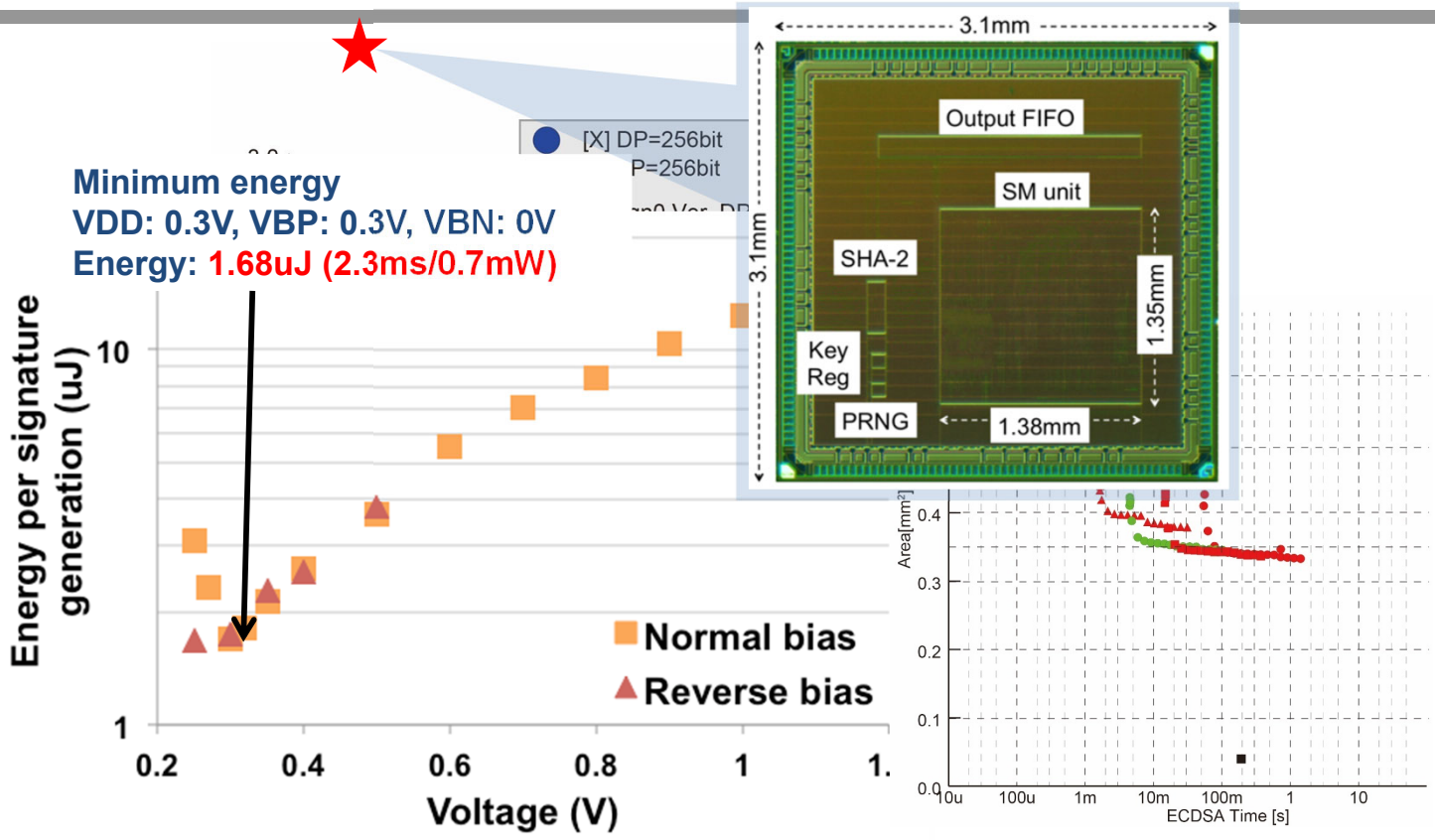


最小遅延：31.3 μs
電源電圧1.4 V
P基板バイアス 0.6 V
N基板バイアス 0.4 V

最小エネルギー：3.28 μJ
電源電圧0.45 V
P基板バイアス 0.2 V
N基板バイアス 0.3 V

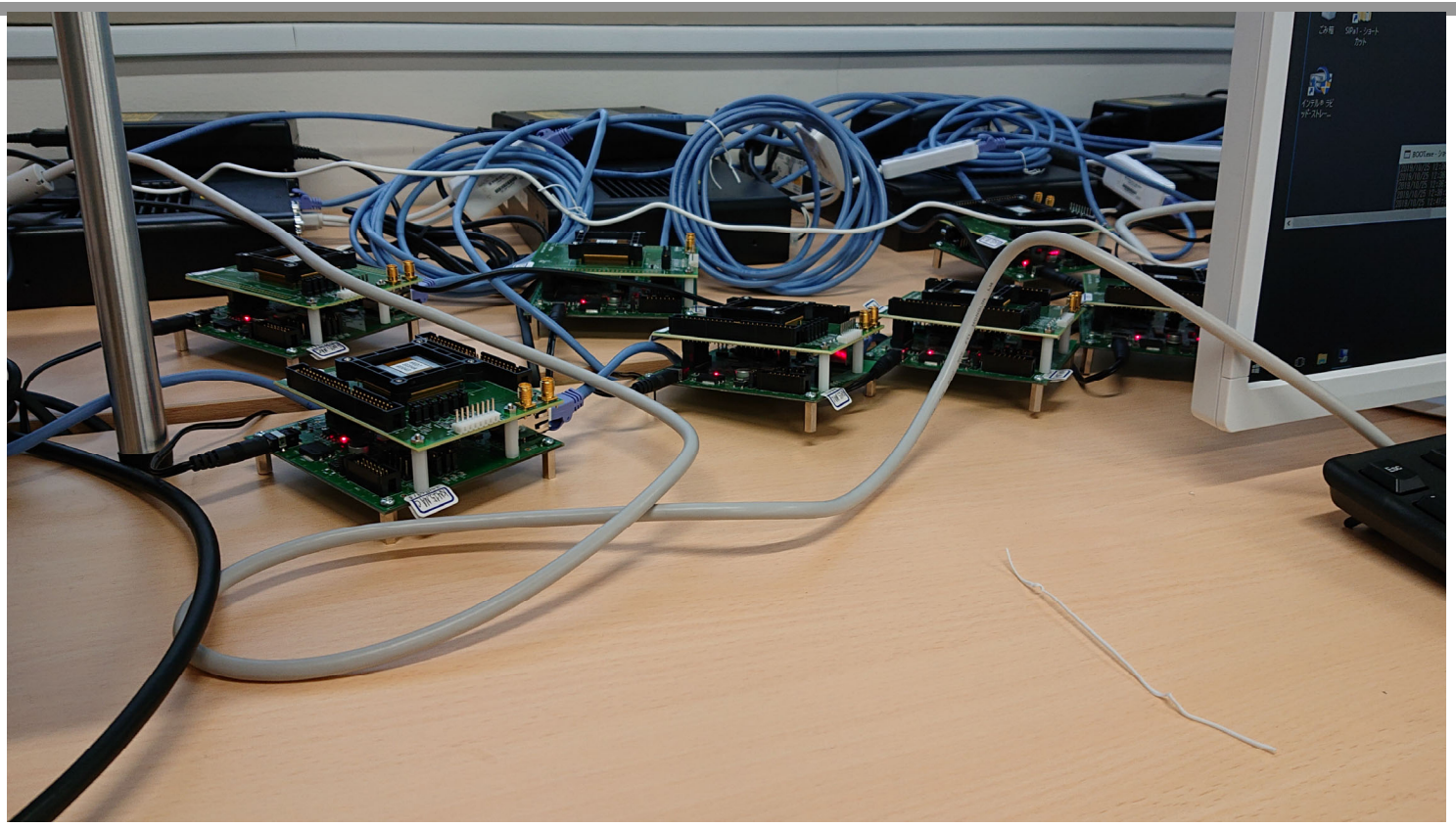
SOTB (Silicon on Thin Buried Oxide) 65nm CMOS

種々の設計例:実測例:最小エネルギー版



SOTB (Silicon on Thin Buried Oxide) 65nm CMOS

NEDOプロジェクトによる実装例



直近の報告との比較

	Plat.	#Gate[kG]	Area[mm^2]	#Clk	Vdd[V]	Freq[MHz]	Tsg[ms]	Pow.[mW]	E[uJ]	Enc/kG	Enc/uJ
Ours	65nm	13	0.03	19.4M	0.75	77	250	**0.16**	100	0.31	0.04
Ours	65nm	1,580	5.64	**7.5k**	0.45	35.7	0.21	15.6	3.28	3.01	**1,452**
					0.75	98.0	0.076	123	9.32	8.33	**1,412**
					1.4	238	**0.031**	1,227	38.7	**20.4**	834
Ours [9]*	65nm	2,500	--	15k	--	236	0.06	--	--	6.67	--
Ours [14]	65nm	1,370	1.92	34.7k	0.25		11	**0.15**	**1.68**	0.07	54.1
					0.3		2.3	0.69	**1.68**	0.32	259
					1.1		0.33	42.9	13.9	2.21	218
[12]*	90nm	540	2.72	22.3k	--	131	0.17	--	--	10.9	--
[15]	Stratix II (90nm)	9,177ALM +96DSP	--	107k	--	157	0.32	--	--	--	--

[9] M. Tamura, IEICE T. Fund. v. 99EA, No. 12, pp. 2444-2452, 2016
[12] S.C. Chung, ISCAS2012, pp. 1456-1459, 2012
[14] M. Tamura, A-SSCC 2016, pp. 341-344, 2016
[15] N. Guillermin, CHES 2010, pp. 48-64, 2010

内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
  - ペアリングエンジンの実装
  - 秘匿検索・属性暗号のペアリングハードウエアによる高速化
  - 準同形暗号とPaillier
- 暗号と安全性
- まとめ

高機能暗号の一例

暗号方式	機能
属性暗号，関数暗号	平文の属性や復号者の属性に合わせてアクセス制御が可能な**可変復号条件暗号**
検索可能暗号	暗号化されたままデータ検索が可能な暗号
代理再暗号化	暗号文の指定復号者の変更に際し暗号文を（平文を経ず）別の暗号文に変換する方式
放送暗号	データ秘匿したまま、多数の指定復号者に対するアクセス制御が可能な暗号
しきい値暗号	しきい値以上のデータにより復元可能な暗号
漏洩耐性暗号	鍵漏えいに対する耐性のある暗号
タイムリリース暗号	時刻による復号制御が可能な暗号

署名方式	機能
属性ベース署名	属性による検証制御ができるディジタル署名
しきい値署名	しきい値以上の個数の署名データにより検証可能となるディジタル署名
グループ署名	ユーザの匿名化ができるディジタル署名
ブラインド署名	署名者に対してメッセージを秘匿するディジタル署名
多重署名，集約署名	署名のセキュアな圧縮、複数の署名文の一括検証が可能なディジタル署名
漏洩耐性署名	鍵漏えいに対する耐性があるディジタル署名
タイムリリース署名	時刻による検証制御が可能なディジタル署名

ペアリングとは・・

- 楕円曲線上で双線形性$e(aP, aQ) = e(P, Q)^{ab}$を有する関数
- IDベース暗号、属性ベース暗号、検索可能暗号などで利用可能
- 計算速度が課題
  - 14nm Intel CPUで 880,000サイクル：210us
  - 20nm FPGAで 18,151サイクル：107us

Optimal Ateペアリングの場合

Input: $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, l = |6u+2| = \sum_{i=1}^{\log_2 l} l_i 2^i$
Output: $a_{opt}(Q,P)$

1: $d \leftarrow g_{Q,Q}(P),\ T \leftarrow 2Q,\ e \leftarrow 1$
2: **if** $l_{\lfloor \log_2 l \rfloor - 1} = 1$ **then** $e \leftarrow g_{T,Q}(P),\ T \leftarrow T + Q$
3: $f \leftarrow d \cdot e$
4: **for** $i = \lfloor \log_2 l \rfloor - 2$ **downto** 0 **do**
5: $\quad f \leftarrow f^2 \cdot g_{T,T}(P),\ T \leftarrow 2T$
6: $\quad$ **if** $l_i = 1$ **then** $f \leftarrow f \cdot g_{T,Q}(P),\ T \leftarrow T + Q$
7: **end for**
8: $Q_1 \leftarrow \phi_p(Q),\ Q_2 \leftarrow \phi_p^2(Q)$
9: **if** $u < 0$ **then** $T \leftarrow -T,\ f \leftarrow f^{p^6}$
10: $d \leftarrow g_{T,Q_1}(P),\ T \leftarrow T + Q_1,$
$\quad e \leftarrow g_{T,-Q_2}(P),\ T \leftarrow T - Q_2,$
$\quad f \leftarrow f \cdot (d \cdot e)$
11: **Final Exponentiaton(FE):**
$\quad f \leftarrow f^{(p^6-1)(p^2+1)(p^4-p^2+1)/r}$
12: **return** f

Miller Loop

$E(\mathbb{F}_{p^2})$の演算と$\mathbb{F}_{p^{12}}$の演算が主な鍵

Final Exponentiation

$\mathbb{F}_{p^{12}}$の演算と逆数演算が主な鍵

Operation	Ops. Count*	Complexity**
Point Doubling	64	76M+50A
Point Addition	6	63M+114A
$\mathbb{F}_{p^{12}}$ Multiplication	350	66M+264A
Frobenius π_p	20	10M+6A

素数位数体F_pと12次拡大体$\mathrm{F}_{p^{12}}$

- $\mathbb{F}_p$: 素数pで割った整数の集合

2nd Ext.

- $\mathbb{F}_{p^2}$: 2乗して-1になる数 i を形式的に追加
  - $a \in \mathbb{F}_{p^2} = a_0 + a_1 i$

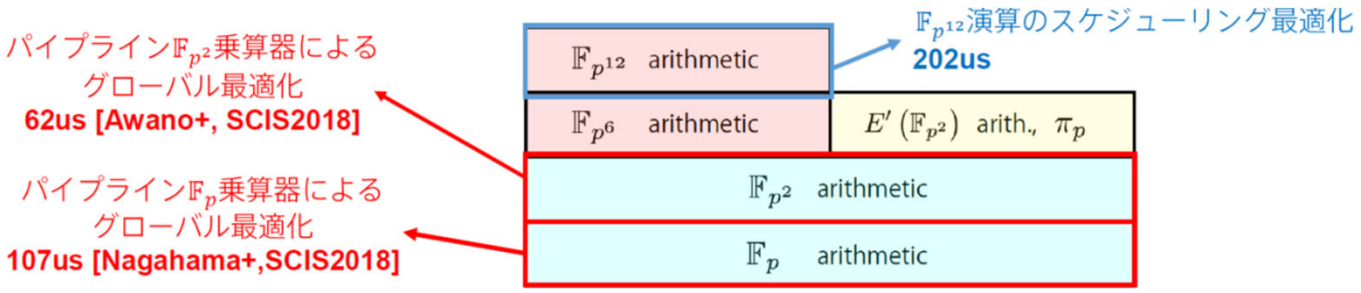
3rd Ext.

- $\mathbb{F}_{p^6}$: 3乗して $\xi = 1 + i$ になる数 v を形式的に追加
  - $a \in \mathbb{F}_{p^6} = a_0 + a_1 v + a_2 v^2 \quad where \quad a_j \in \mathbb{F}_{p^2}$
    $= (a_{00} + a_{01} i) + (a_{10} + a_{11} i) v + (a_{20} + a_{21} i) v^2 \; where \quad a_{jk} \in \mathbb{F}_p$

2nd Ext.

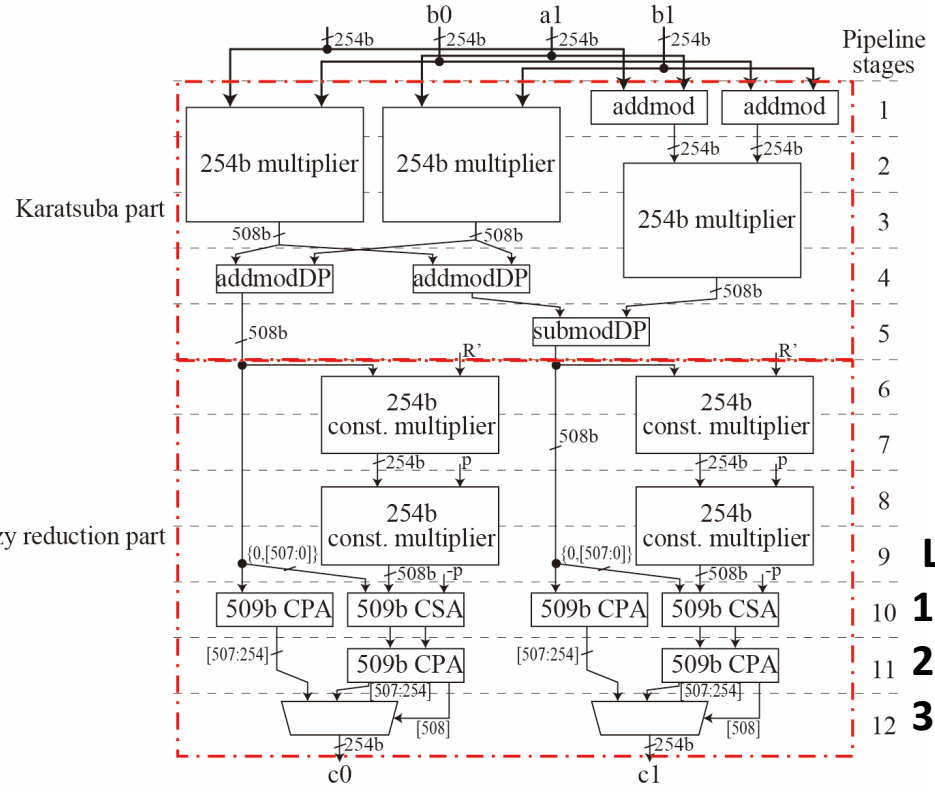
- $\mathbb{F}_{p^{12}}$: 2乗して v になる数 w を形式的に追加
  - $a \in \mathbb{F}_{p^{12}} = a_0 + a_1 w = (a_{00} + a_{01} v + a_{02} v^2) + (a_{10} + a_{11} v + a_{12} v^2) w$

12次拡大体$\mathrm{F}_{p^{12}}$とFEの実装



Name	サイクル	充足率
Miller Loop	3,697	99.3%
Final Addition	162	78.8%
Final Exponentiation Easy Part	431	25.3%
Final Exponentiation Hard Part	3,885	88.4%
Pairing	8,175	84.6%

12段パイプライン構成の2次拡大体乗算器



Karatsuba method:

$\gamma = \alpha \times \beta$:

$\alpha \in \mathrm{Fp}_2 = a_0 + a_1 i,$

$\beta \in \mathrm{Fp}_2 = b_0 + b_1 i,$

$\gamma \in \mathrm{Fp}_2 = c_0 + c_1 i$ **as**

$c_0 = a_0 b_0 - a_1 b_1, c_1$

$= [(a_0 + a_1)(b_0 + b_1)$

$-a_0 b_0 - a_1 b_1]$

Lazy Reduction:

1) Multiplication

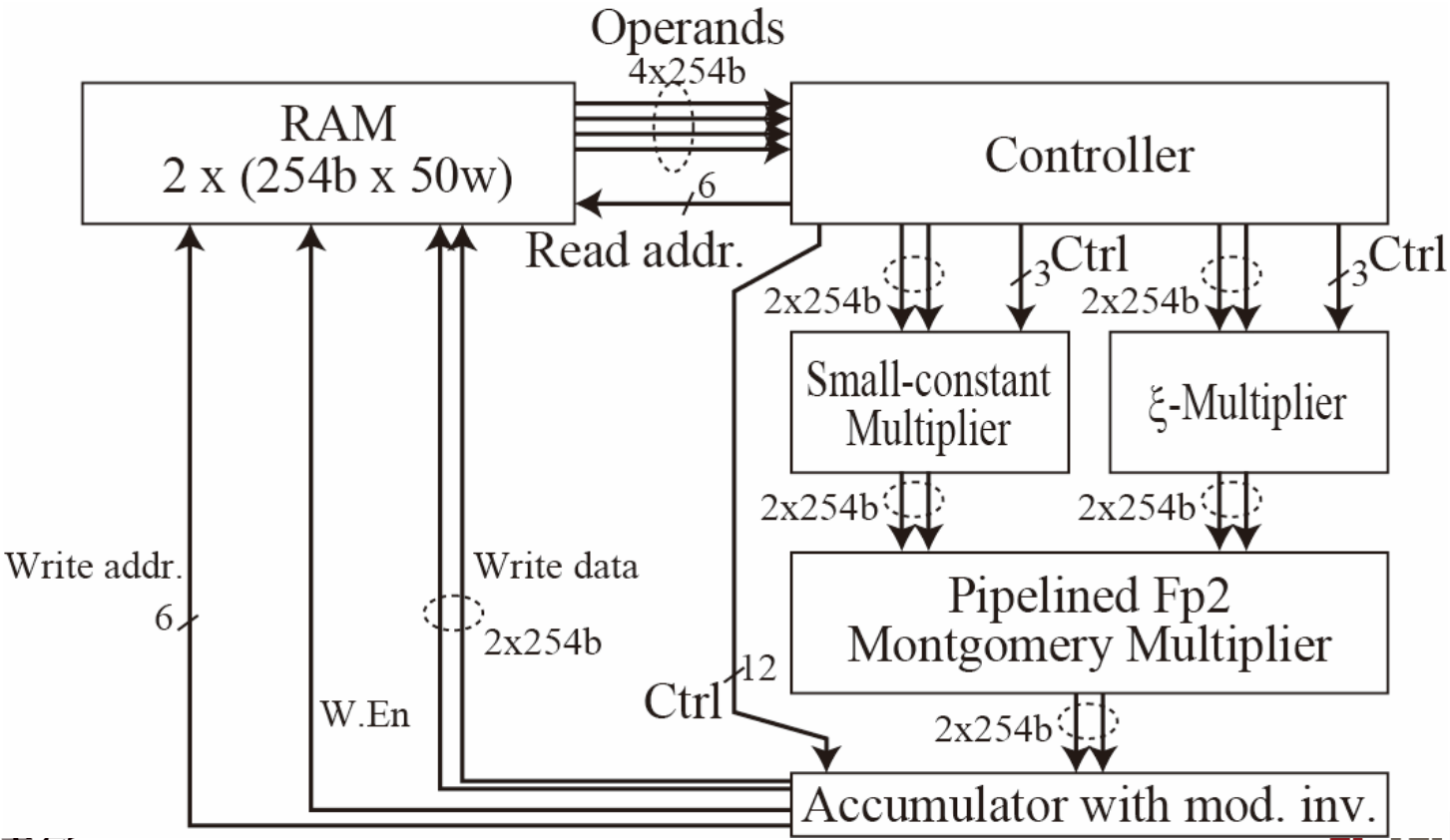
2) addition/subtraction in double precision

3) Reduction to single precision

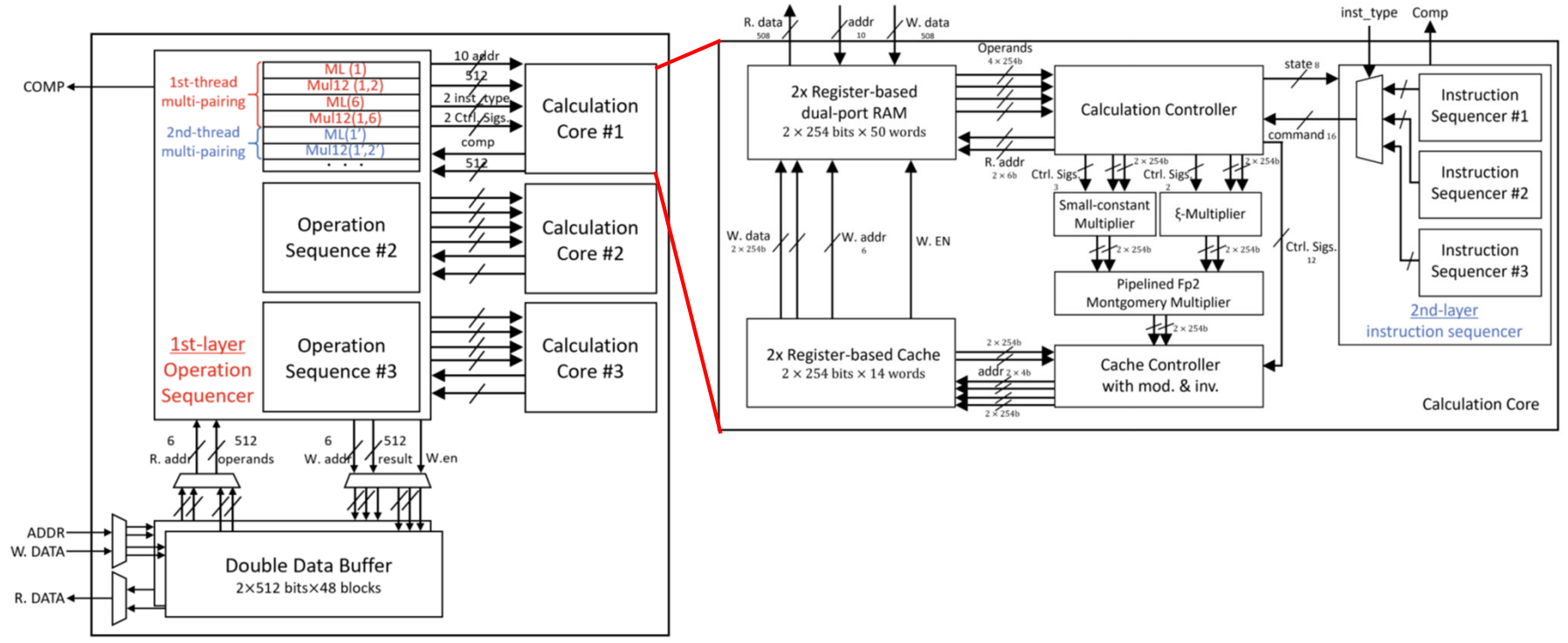
スケジューリング最適化

Name	Operation	#ops	Cycles/op	Operation efficiency
Init-PADD	Initialize – S12 – PDBL – SM12 – PADD – SM12	1	93	72.0%
PADD	S12 – PDBL – SM12 – PADD – SM12	4	107	90.7%
PDBL	S12 – PDBL – SM12	58	56	100%
Final Addition	PADD – PADD - SM12 – PADD – SM12	1	162	78.8%
FE-EP	Final Exp. Easy Part incl. Inv.	1	431(254)	25.3%
SQR_{012345}	S12 by SQR	192	17	76.4%
M12	Mult. in Fp_{12}	20	39	92.3%
Frobenius	1^{st} and 2^{nd} Frobenius	7	12	58.3%
CONJ12	Conjugate in Fp_{12}	2	12	50.0%

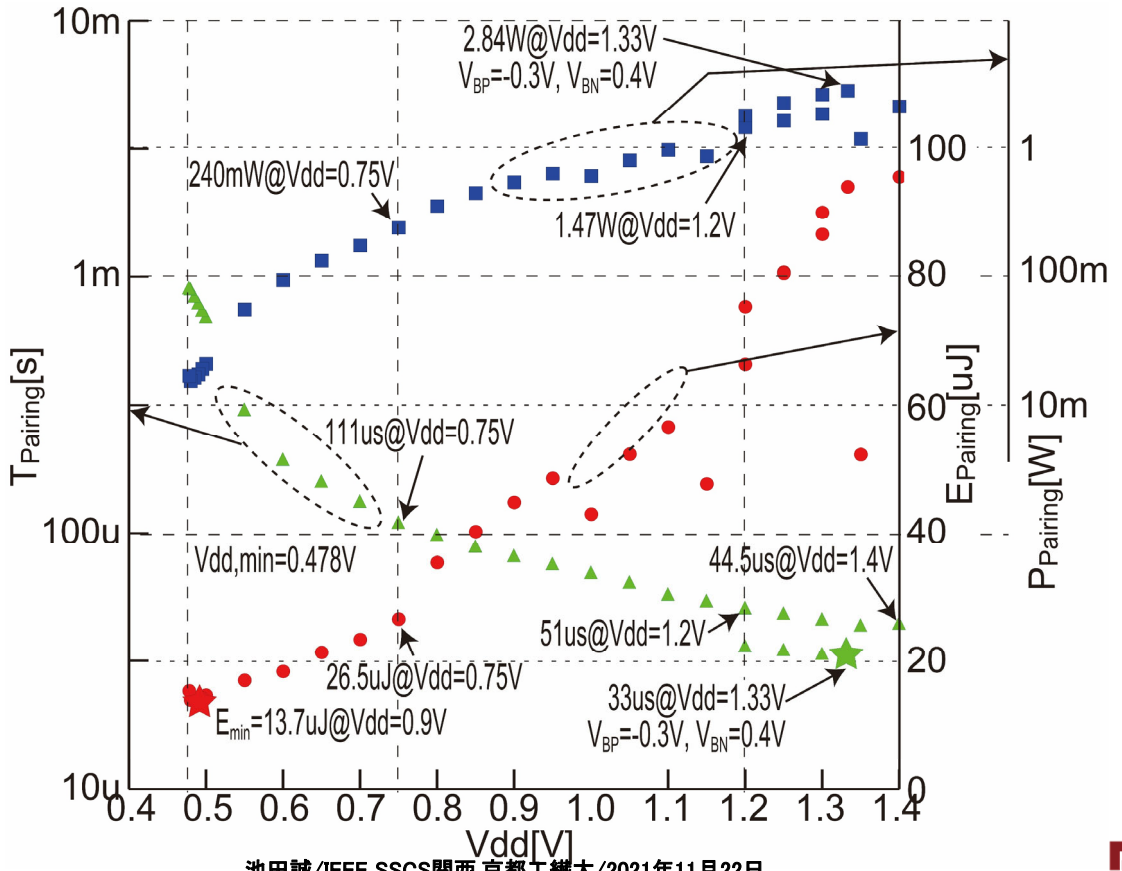
チップ全体構成(Fp_{12} 部)



複数コア構成のチップ全体



チップの測定結果


2.84W@Vdd=1.33V
V_{BP}=-0.3V, V_{BN}=0.4V
240mW@Vdd=0.75V
1.47W@Vdd=1.2V
111us@Vdd=0.75V
Vdd,min=0.478V
44.5us@Vdd=1.4V
51us@Vdd=1.2V
26.5uJ@Vdd=0.75V
E_{min}=13.7uJ@Vdd=0.9V
33us@Vdd=1.33V
V_{BP}=-0.3V, V_{BN}=0.4V
$T_{Pairing}$[s]
$E_{Pairing}$[uJ]
$P_{Pairing}$[W]
Vdd[V]
10m
1m
100u
10u
100
80
60
40
20
0
1
100m
10m
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4

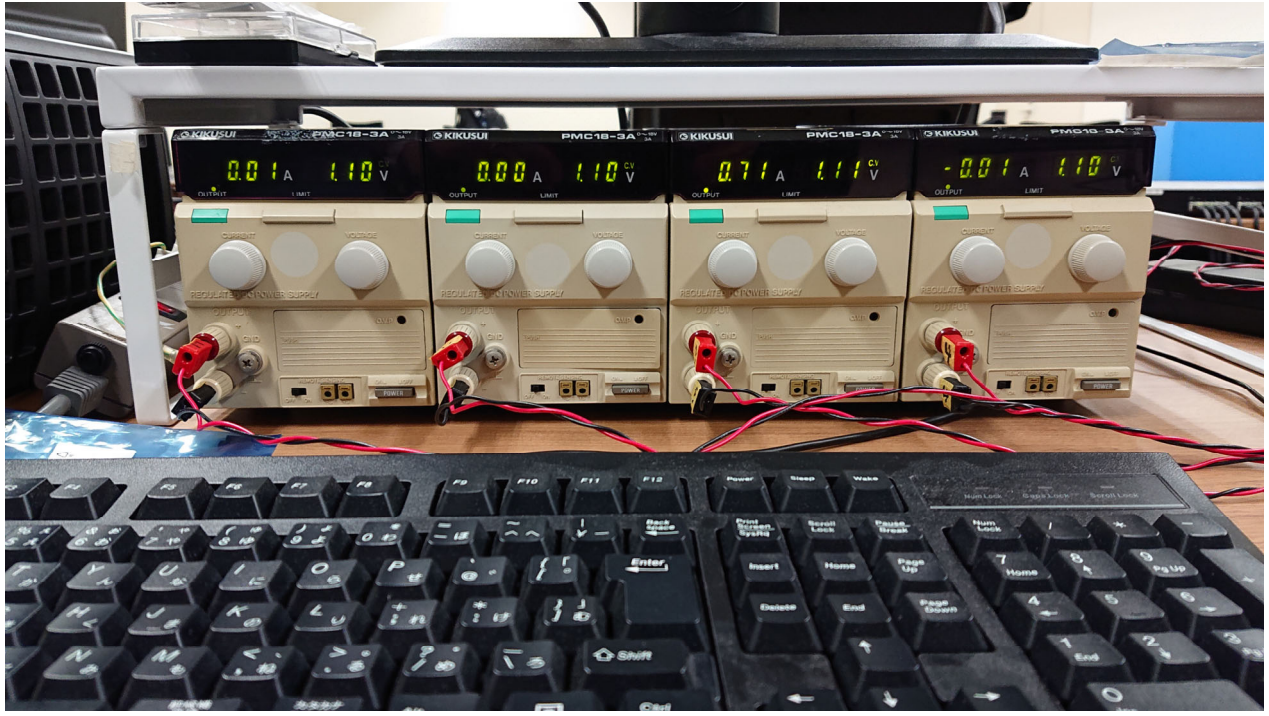
BN254ペアリング実装の比較

Platform	#Gates [kG]	Area [mm^2]	Vdd[V]	Freq [MHz]	1-pair				7-pair (*est / # measure)					
					#Clk	t_{Pair} [us]	P_{Pair} [mW]	E_{pair}[uJ]	Vdd	Freq	#Clk	t_{Pair} [us]	P_{Pair} [mW]	E_{pair}[uJ]
Mobile Device[19] Apple A5 32nm	-	-	-	1,000	9,909,000	9,905	-	-			38,139,741	38,140	-	-
HighendPC[20] Corei7-6700K 14nm	-	-	-	4,000	840,000	210	91,000	19,110			3,233,160	808	91,000	12,444,433
HighendFPGA[13] KintexUltra 20nm	14,463slice+460DSP	-	-	170	18,151	107	-	-			69,863	412	-	-
ASIC*[21] 130nm	94	-	-	338	5,340,400	15,800	-	-			20,555,200	60,814	-	-
ASIC[22] 65nm	354	2.51	-	800	512,541	640	255	163			1,972,770	2,463	255	627
ASIC**[23] 65nm	323	-	-	633	330,053	521	-	-			1,270,374	2,005	-	-
ASIC*[14] 65nm	3,205	12.8		147	9,270	202					35,680	777		
ASIC [2] 65nm FDSOI	2,793	12.8	1.33	250	8,175	33	2,850	94.0	1.33	250	31,466	127	2,850	361.8
			0.75	74.6		110	240	26.5	0.75	74.6		287	240	102.0
			0.49	9.2		792	17.2	13.7	0.49	9.2		3,048	17.2	52.7
ASIC[12] 65nm FDSOI	13,104	21.62	1.4	105	8,000	76.0	1,188	90.3	1.4	66.7	13,784	206.8	1,175	243.0
			0.75	47.6		168	68.2	11.5	0.75	29.9		461.8	55.4	25.6
			0.32	2.33		3,440	0.95	3.28	0.32	2.13		6,479	1.02	6.62

内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
  - ペアリングエンジンの実装
  - 秘匿検索・属性暗号のペアリングハードウエアによる高速化
  - 準同形暗号とPaillier
- 暗号と安全性
- まとめ

秘匿検索の高速化（NEDOプロジェクト）



属性暗号の実装

Set up

$$MPK = \{ P, Q, P_\delta = \delta P, \gamma = e(P,Q)^\alpha \}, MSK = \{ P_\alpha = \alpha P \}.$$

Encryption

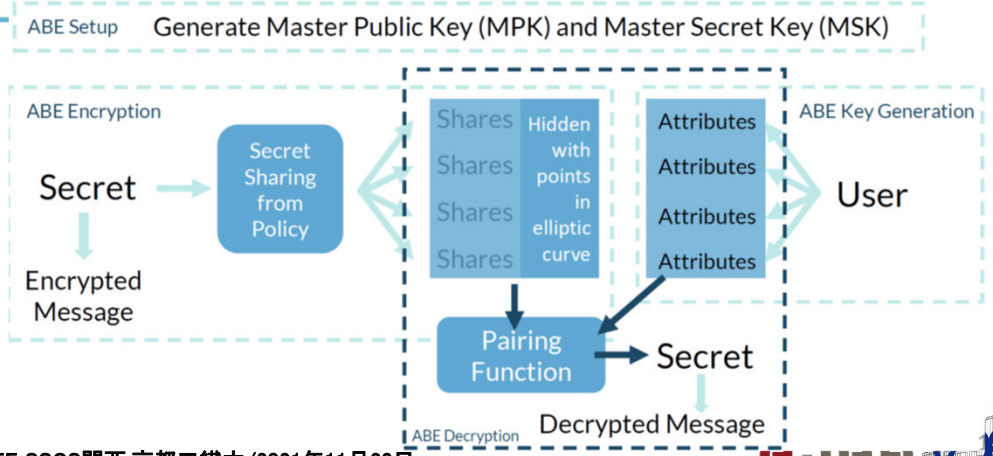
$$CT = \{ Policy, C = M + \gamma^s, C_d = sQ, \forall i \in \boldsymbol{I}: C_i = \lambda_i P_\delta - x_i \mathcal{H}_i, D_i = x_i Q \}$$

Keygen

$$SK = \{ K = P_\alpha + \tau P_\delta, L = \tau Q, \forall i \in \boldsymbol{T}: K_i = \tau \mathcal{H}_i \}$$

Decryption

$$\gamma^s = \left[e(-\Delta K, C_d) e\left(\sum w_i C_i, L \right) \prod e(w_i K_i, D_i) \right]^{1/\Delta}$$



属性暗号のハードウエアによる高速化

Steps	Operation	Time (us)	Estimated Time (us)
Setup	G1 Rand	815.7	**140.0**
	G1 Map	1,182.3	1,182.3
	G1 Mul	13,940.3	**280.0**
	G2 Rand	1,967.0	**70.0**
	G2 Mul	24,413.7	**140.0**
	GT Exp	5,333.3	5,333.3
	Pairing	9,652.3	**80.0**
	File Read/Write	2,639.7	2,639.7
	File AES Enc	1,034.0	1,034.0
	Others	1,675.3	1,675.3
	Total	62,653.7	12,441.2
		Speed Up	***5.04***
User Registration	G1 Map	436.3	436.3
	G1 Mul	3,964.7	**70.0**
	G2 Mul	5,592.0	**35.0**
	File Read/Write	893.0	893.0
	File AES Enc	1,229.0	1,229.0
	Others	745.0	745.0
	Total	12,860.0	3,408.3
		Speed Up	***3.77***

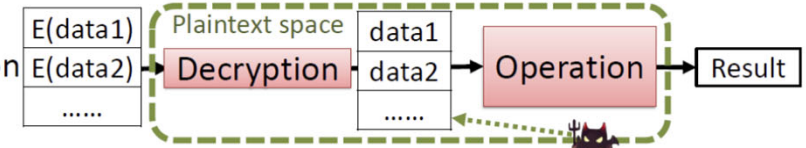
Steps	Operation	Time (us)	Estimated Time (us)
Encryption	G1 Map	4,348.3	4,348.3
	G1 Mul	33,018.0	**700.0**
	G2 Mul	52,872.0	**385.0**
	GT Exp	3,065.3	3,065.3
	File Read/Write	1,569.7	1,569.7
	Others	920.3	920.3
	Total	95,793.7	10,988.7
		Speed Up	***8.72***
Decryption	G1 Map	3,805.3	3,805.3
	G1 Mul	41,926.3	**1,085.0**
	G2 Mul	4,529.7	**35.0**
	Pairing	70,384.7	**720.0**
	File Read/Write	2,231.0	2,231.0
	Others	2,903.3	2,903.3
	Total	125,780.3	10,753.3
		Speed Up	***11.70***

内容

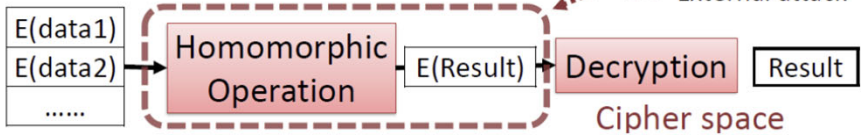
- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
  - ペアリングエンジンの実装
  - 秘匿検索・属性暗号のペアリングハードウエアによる高速化
  - 準同形暗号とPaillier
- 暗号と安全性
- まとめ

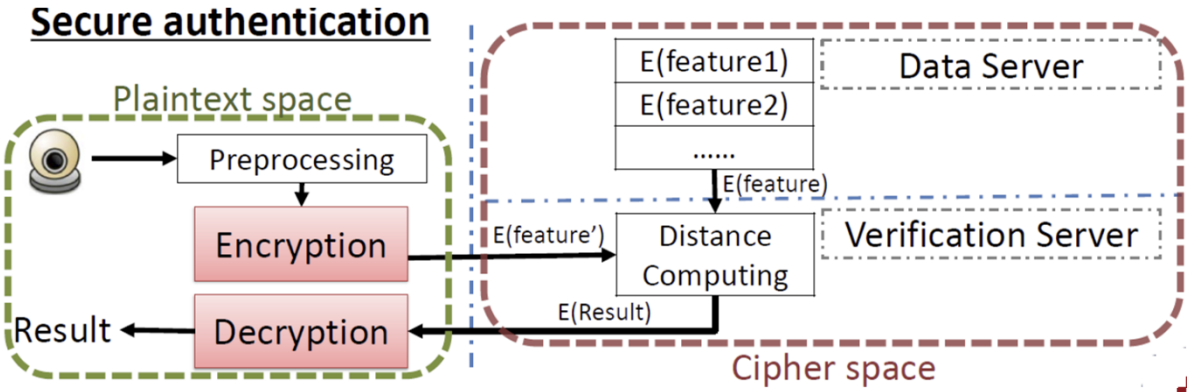
準同形暗号

- Conventional encryption method



- For homomorphic encryption

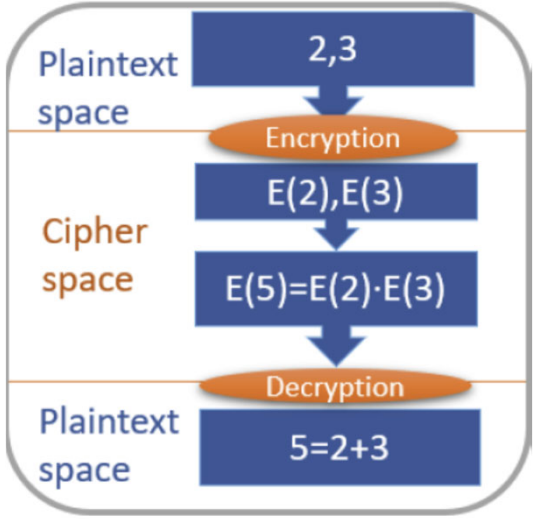




準同形暗号

Type	Supported Operations	Representative Encryption method
加法準同性	+	Paillier Encryption
乗法準同性	X	RSA Encryption
完全準同性	+, X	Ring-LWE/Torus FHE

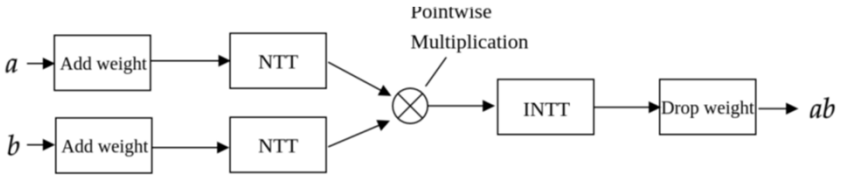
加法準同性

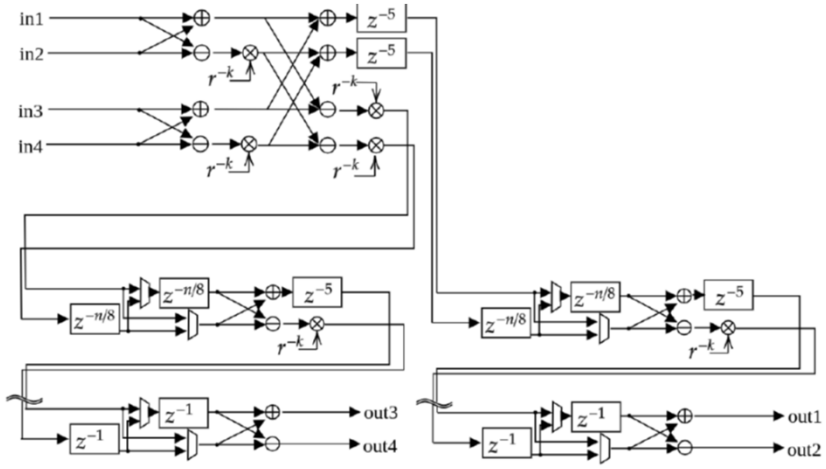


RingLWE (数論変換NTT)の実装

$$A \cdot s + e \equiv x \pmod{q}$$

行列Aおよび対象ベクトルxが与えられるとき、誤差eを付与したときにsを求める問題

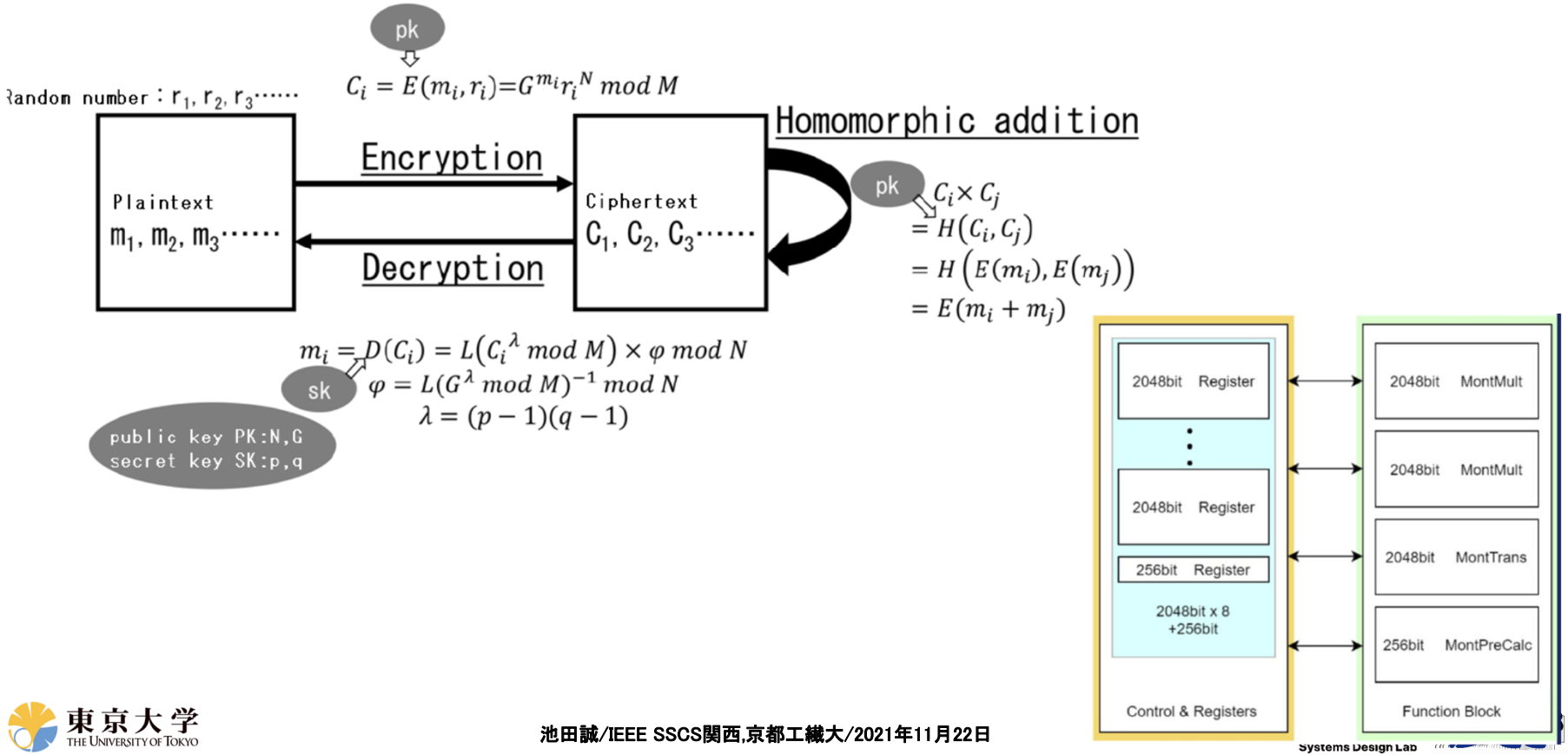




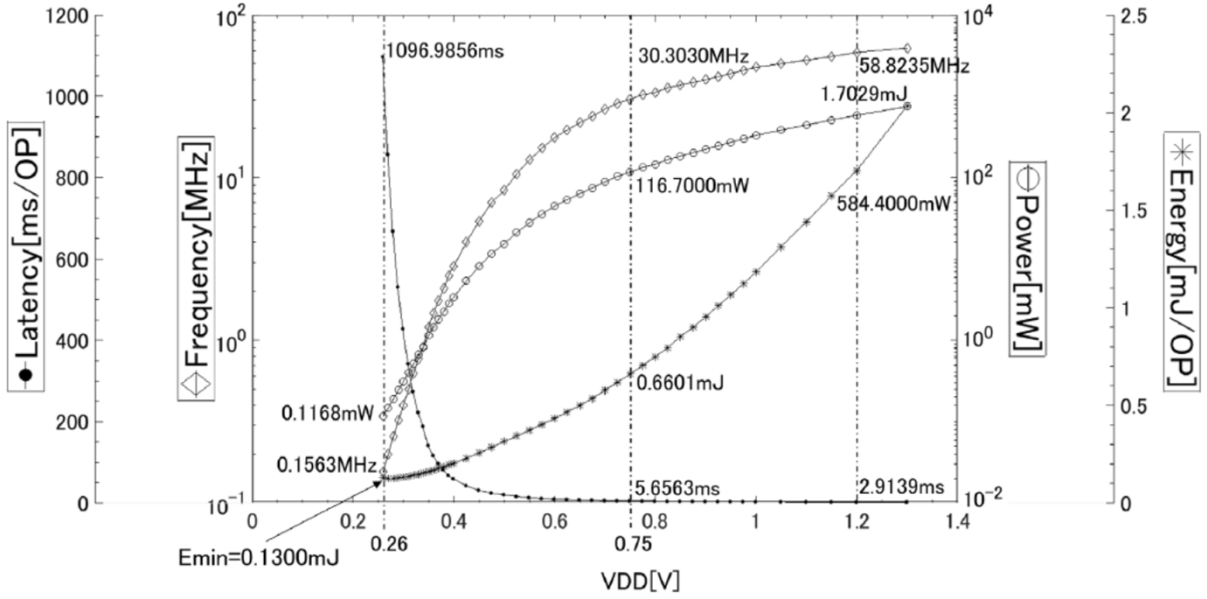
Design	Platform	Resource	n	q	Clock (MHz)	Latency (μs) NTT	Latency (μs) Pol. Mul.	Throughput(1000op/s) NTT	Throughput(1000op/s) Pol. Mul.
Du (2016) [18]	SPARTAN-6	1246 LUT / 16 DSP / 6 BRAM	512	23-bit	249	-	6.78	-	147
Feng (2020) [19]	SPARTAN-6	18k LUT / 128 DSP / 2.5 BRAM	512	23-bit	233	-	1.77	-	565
Mert (2020) [20]	VIRTEX-7	77k LUT / 952 DSP / 325.5 BRAM	1024	32-bit	200	0.4	0.96	2500	1041
Banerjee (2019) [11]	40nm CMOS	106kGates [a]	256	24-bit	72	17	-	59	-
Song (2018) [13]	40nm CMOS	2.05mm^2 [a]	512	18-bit	300	1.64	-	609	-
This Work (2-stream)	**65nm CMOS**	**571kGates**	**512** (1024)	**23-bit**	**970**	**0.544** (1.09)[b]	**0.840** (1.68)[b]	**3792** (1896)[b]	**3792** (1896)[b]
This Work (32-stsream)	**65nm CMOS**	**5405kGates**	**512** (1024)	**23-bit**	**806**	**0.060** (0.120)[b]	**0.118** (0.236)[b]	**50403** (25201)[b]	**50403** (25201)[b]

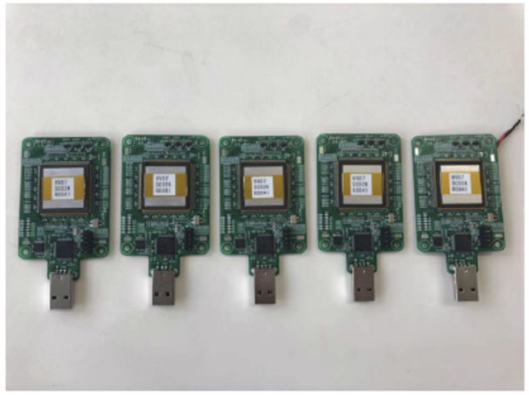
[a]Contain other modules for Ring-LWE [b]Estimated latency/throughput for $n = 1024$

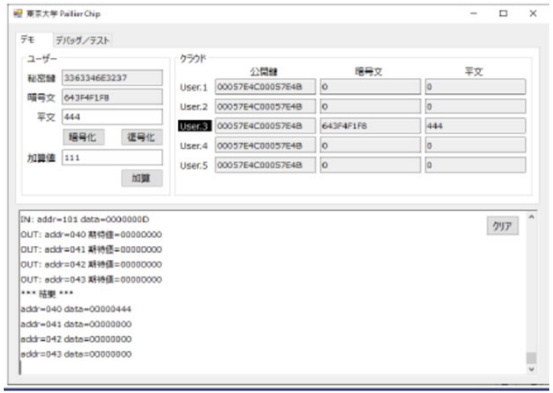
Paillierアルゴリズムとハードウエア構成



Paillierチップ測定結果とデモ







SOTB (Silicon on Thin Buried Oxide) 65nm CMOS

内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
  - ファクタリング・離散対数問題・楕円曲線上離散対数問題
  - 量子計算と耐量子計算暗号
  - 暗号アルゴリズムと対タンパ性
    - アルゴリズム的理解
    - 回路的理解
- まとめ

利用推奨暗号

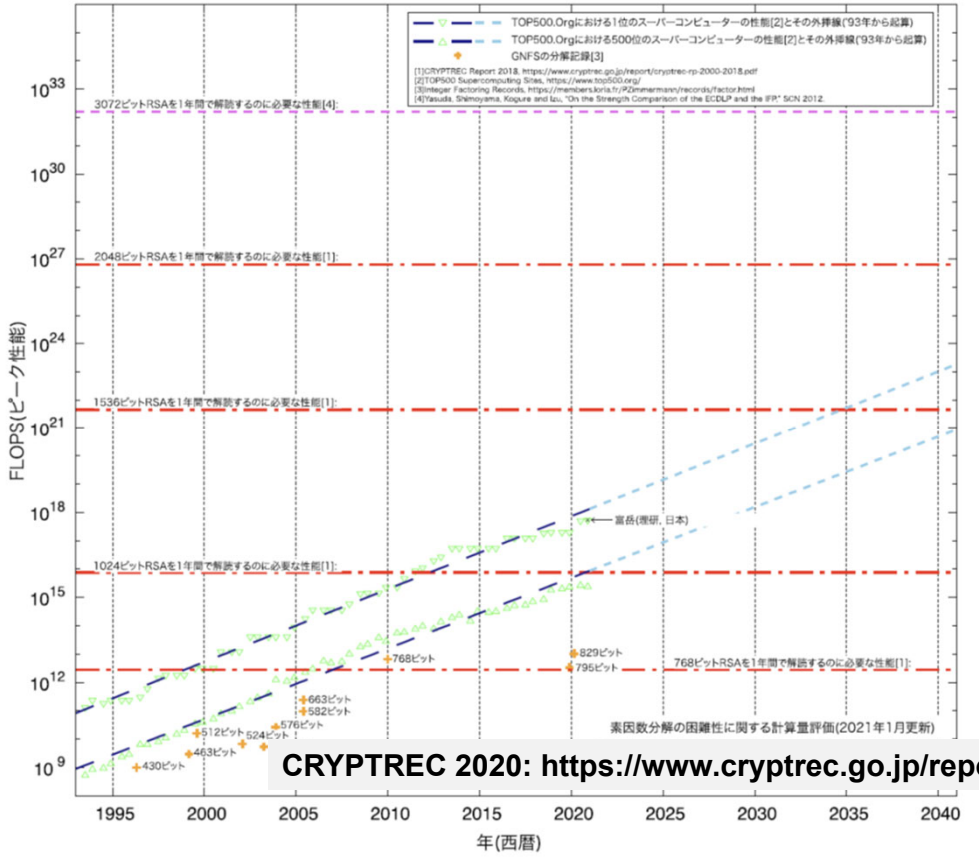
		高セキュリティ型	推奨セキュリティ型	セキュリティ例外型	共通の利用禁止暗号
鍵交換		DHE ECDHE	DHE ECDHE	DH DHE ECDH ECDHE RSAES-PKCS1-V1_5	なし
署名		ECDSA RSASSA-PKCS1-v1_5 RSASSA-PSS*2*3	ECDSA RSASSA-PKCS1-v1_5	ECDSA RSASSA-PKCS1-v1_5	GOST R 34.10-2012*1
暗号化	64ビットブロック暗号				RC2, EXPORT-RC2 IDEA DES, EXPORT-DES GOST 28147-89*1 Magma*1 3-key Triple DES
	128ビットブロック暗号	AES Camellia*4	AES Camellia*4	AES Camellia*4	Kuznyechik*1 ARIA SEED
	利用モード	CCM CCM_8 GCM	CBC CCM CCM_8 GCM	CBC CCM CCM_8 GCM	CTR_OMAC*1
	ストリーム	ChaCha20-Poly1305	ChaCha20-Poly1305	ChaCha20-Poly1305	RC4, EXPORT-RC4
ハッシュ関数(HMAC)		SHA-256 SHA-384	SHA-1 SHA-256 SHA-384	SHA-1 SHA-256 SHA-384	MD5 GOST R 34.11-2012*1
推奨しないが利用を妨げない		DSA EdDSA*2	RSAES-PKCS1-V1_5 DSA EdDSA*2	DSA EdDSA*2	凡例: *1 現在ID(RFCではない) *2 暗号スイートのIANA登録番号として管理されていない *3 TLS1.3でのみ利用可 *4 TLS1.2以前で利用可
追加の利用禁止暗号		DH ECDH RSAES-PKCS1-V1_5 CBC SHA-1	DH ECDH	なし	

CRYPTREC 2020: https://www.cryptrec.go.jp/report/cryptrec-mt-1011-2020.pdf

セキュリティレベル

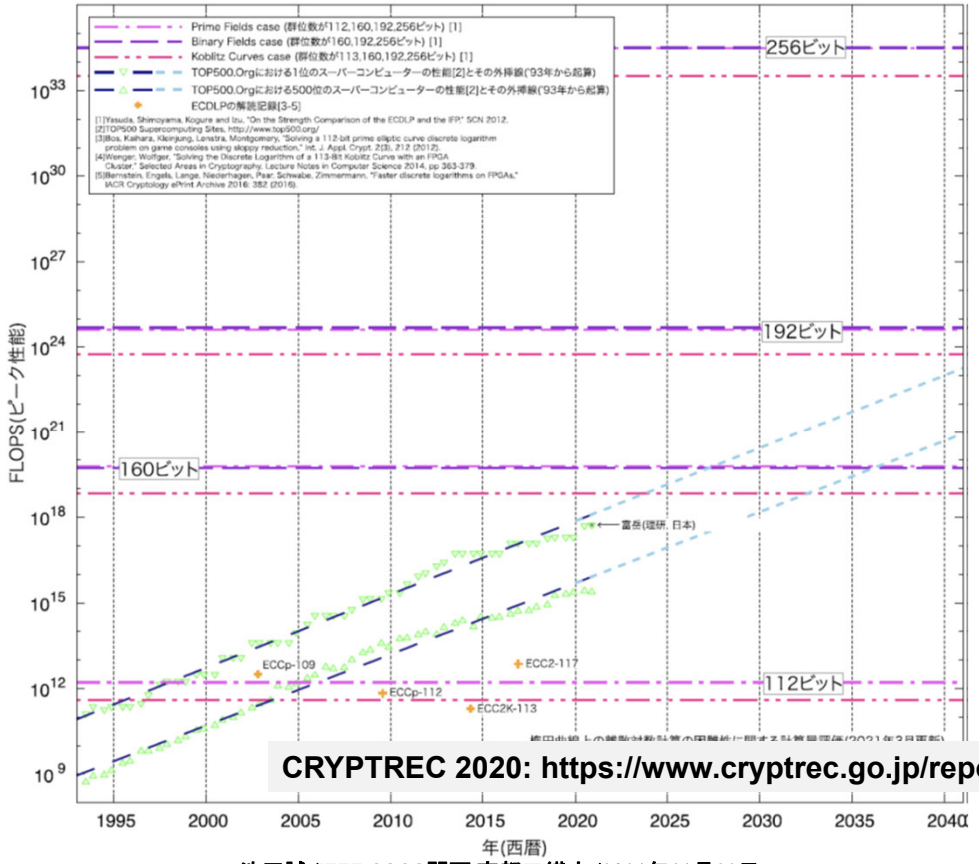
- RSA
  - Factoring large numbers
  - RSA problem
- ECDSA
  - ECDLP (elliptic curve discrete logarithm problem)
- Pairing
  - ECDLP
  - DLP (discrete logarithm problem)

暗号の解読時間（RSA）



CRYPTREC 2020: https://www.cryptrec.go.jp/report/cryptrec-mt-1021-2020.pdf

暗号の解読時間（ECDLP）



CRYPTREC 2020: https://www.cryptrec.go.jp/report/cryptrec-mt-1021-2020.pdf

ペアリング曲線の選択と安全性

Curve	Embedded degree	Key length [bit]	Database size [bit]	Security level [bit]
BN Curve	12	224	2,688	112 → ???
		254	3,048	128 → 100~110
		256	3,072	128
		512	6,144	128
BLS Curve	12	381	4,572	128
	24	381	9,144	192
KSS Curve	18	384	6,912	192
FourQ	3.7×10^{73}	256	9.5×10^{75}	???

DLP and ECDPL

- DLP
  - NFS (Number Field Sieve)

$$L_p(s,c) = \exp\left((c + o(1))(\log p)^s(\log\log p)^{1-s}\right)$$

p: Characteristic

  - S.ex.TNFS (Special Extended Tower NFS)

$$L_p(1/3, (32/9)^{1/3})$$

- ECDLP
  - Pollard ρ-argorithm $O(\sqrt{r})\left(= \exp\left(\frac{1}{2}(\log r)^1\right)\right)$

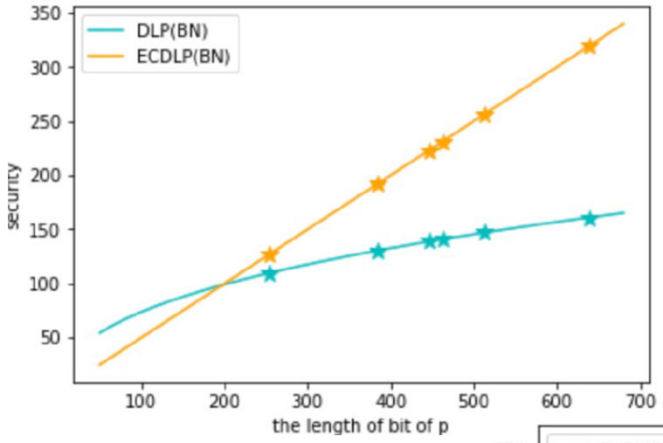
ペアリングのパラメータとセキュリティレベル

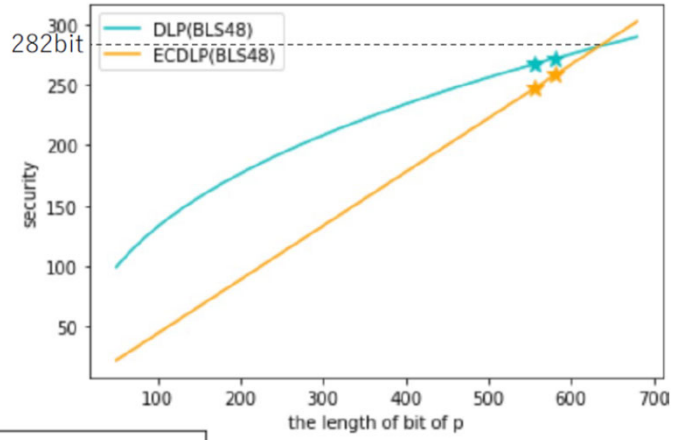
BN Curve	BN254	BN384	BN446	BN462	BN512	BN638
Size of Q	254*12=3048	384*12=4608	446*12=5352	462*12=5544	512*12=6144	638*12=7656
ECDLP	127	192	223	231	256	319
DLP	110	131	139	141	147	161
Security Level	110	131	139	141	147	161

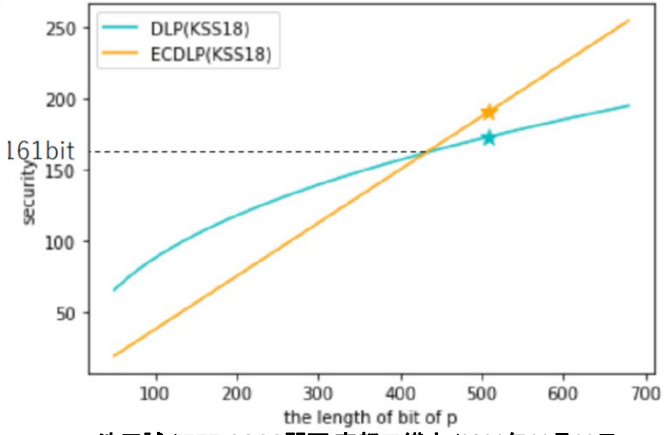
BLS12	BLS12-381	BLS12-383	BLS12-446	BLS12-455	BLS12-477	BLS12-575	BLS12-638
Size of Q	4572	4596	5352	5460	5724	6900	7656
ECDLP	127	128	149	151	159	192	213
DLP	131	131	139	140	143	155	161
Security Level	127	128	139	140	143	155	161

	BLS24-479	BLS48-556	BLS48-581	KSS18-508
Size of Q	11496	26688	27888	9144
ECDLP	192	248	259	191
DLP	191	268	272	174
Security Level	191	248	259	174

ペアリングのパラメータとセキュリティレベル







内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
  - ファクタリング・離散対数問題・楕円曲線上離散対数問題
  - 量子計算と耐量子計算暗号
  - 暗号アルゴリズムと対タンパ性
    - アルゴリズム的理解
    - 回路的理解
- まとめ

量子コンピューター「3年で実現する」

IBMクリシュナCEO、世界デジタルサミットで

2021年6月8日 14:30 [有料会員限定]

保存

人工知能（AI）や高速通信規格「5G」を生かしたデジタル技術の革新について議論する「世界デジタルサミット2021」（日本経済新聞社・総務省主催）は8日、2日目の討議に入った。米IBMのアービンド・クリシュナ会長兼最高経営責任者（CEO）は対談形式の講演で、次世代の高速計算機である量子コンピューターについて「3年ぐらいで実現する」と述べ、人工知能（AI）の普及を後押しすると強調した。

オンラインで参加した米IBMのクリシュナ会長兼CEO（8日午前）

IBMは米グーグルと共に世界の量子コンピューターの研究開発をけん引し、日本では東京大学などと連携している。クリシュナCEOは「現在の量子コンピューターは既存のコンピューターと同様のことができる程度だが、2023年をメドに（現在よりも大幅に計算能力の高い）1000量子ビット超の性能を開発する」と語った。

量子コンピューターの発展は多様な産業でAIの導入やデジタルトランスフォーメーション（DX）を加速するため「30年までにAIが全世界に及ぼす経済効果は16兆ドル（約1750兆円）に迫る」との見通しも示した。

顧客が社内外のデータセンターを組み合わせて利用でき、システム投資を抑えることができる「ハイブリッドクラウド」の技術もDXのなかで重要な技術になると話した。IBMはこのハイブリッドクラウド事業を自社の成長分野としている。

日経新聞電子版2021/6/8 14:30配信

量子計算機と暗号解読

- 古典計算機で1万年かかる計算が量子計算機では200秒で完了する
  [Quantum supremacy using a programmable superconducting processor, https://www.nature.com/articles/s41586-019-1666-5]
  - この論文では、53量子ビット：1543量子ゲート
  - １演算1us
  - 誤りなしの仮定
  - この時RSA2048の素因数分解には
  - 4096量子ビット：10^{12}~10^{13}量子ゲート
  - 誤り訂正が必要な場合2000万量子ビット必要

→当面量子計算により解読されるリスクは低い、ただし今後の技術革新には十分注視が必要

→（念のため）耐量子計算暗号の標準化も進んでいる

耐量子計算暗号(NIST PQE)

- 格子暗号(RingLWE)
- 楕円曲線間の同種写像(SIKE)

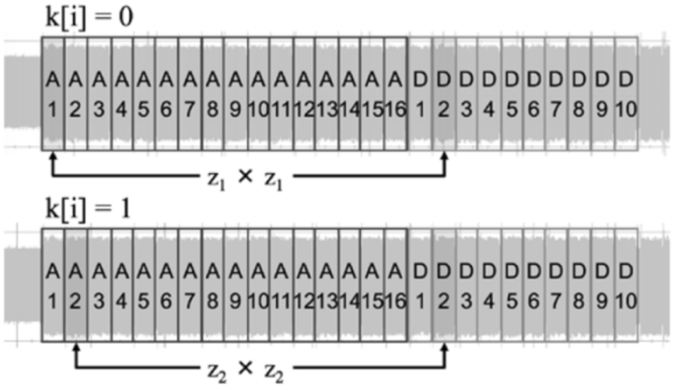
内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
  - ファクタリング・離散対数問題・楕円曲線上離散対数問題
  - 量子計算と耐量子計算暗号
  - 暗号アルゴリズムと対タンパ性
    - アルゴリズム的理解
    - 回路的理解
- まとめ

暗号エンジンと耐タンパー性

- アルゴリズム / アーキテクチャ
  - モンゴメリラダー構成
- データタイミング
  - 演算子置換
  - 演算置換
- 回路
  - 非同期制御 / 二線式回路

アルゴリズムレベルでの耐タンパー性




```
if k[i]=0 then
   P2 <= P1 + P2
   P1 <= 2P1
else
   P1 <= P1 + P2
   P2 <= 2P2
endif
```

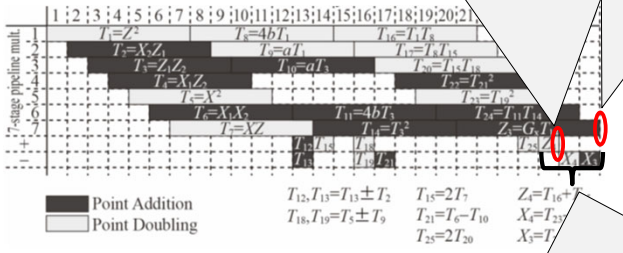
k[i]=0:
Operations A1 and D2 are identical

k[i]=1:
Operations A2 and D2 are identical

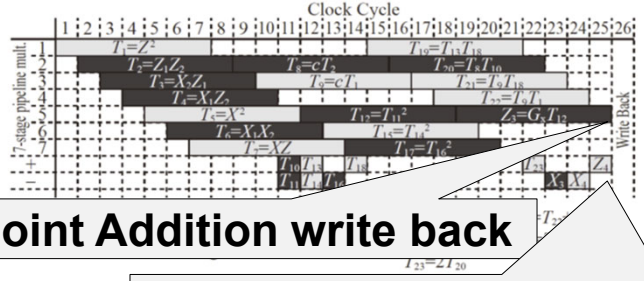
```
if k[i]=0 then
   P2 <= P1 + P2
   P1 <= 2P1
else
   P1 <= P2 + P1
   P2 <= 2P2
endif
```

Point Addition write back

Point Doubling write back



Write-back timing reveals secret information k[i]

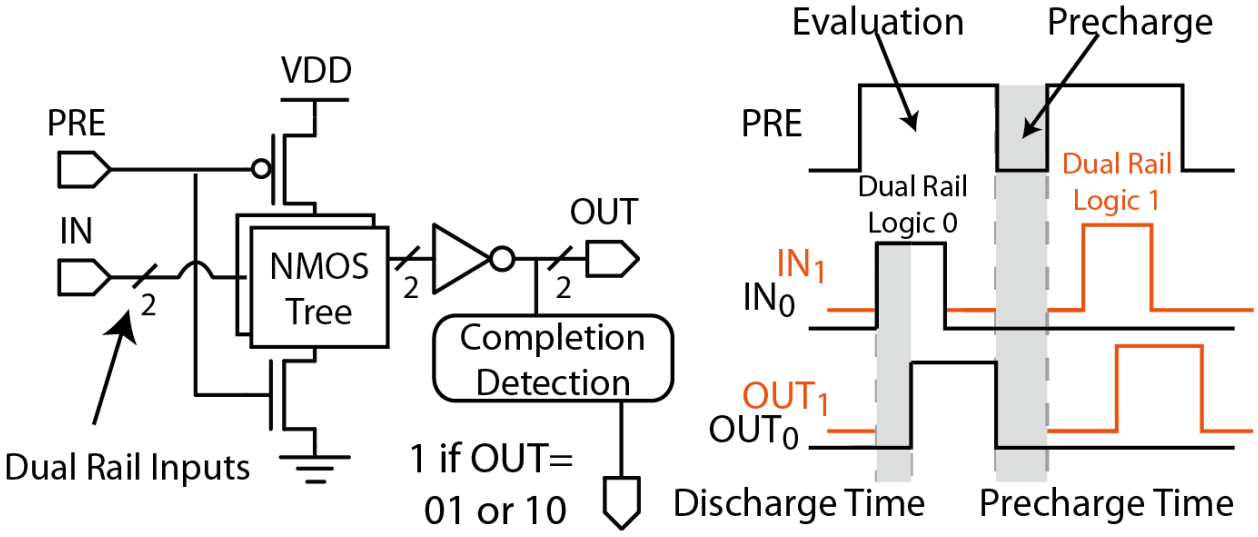


Point Addition write back

Point Doubling write back

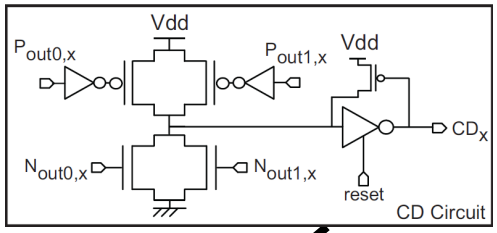
二線 / 非同期制御

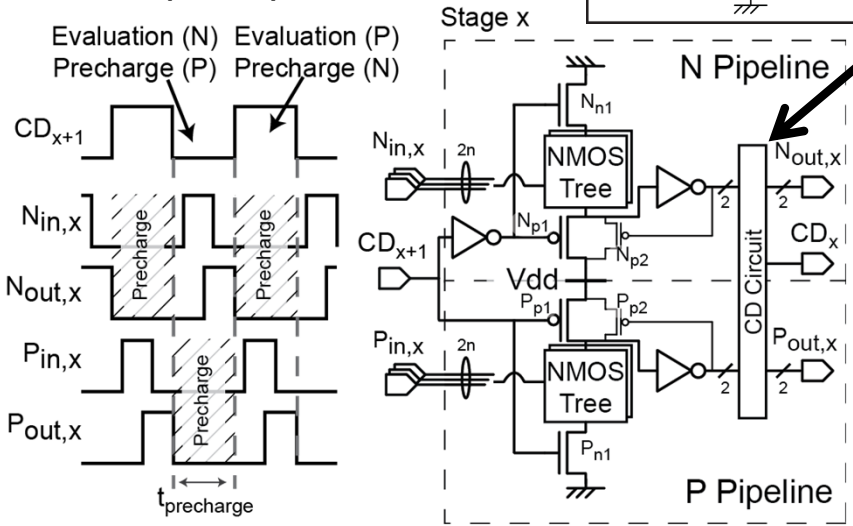
- **Dynamic circuits (DCVSL) & Dual-rail logic**
  - **Always one transition per one cycle**
  - **Require pre-charge control**



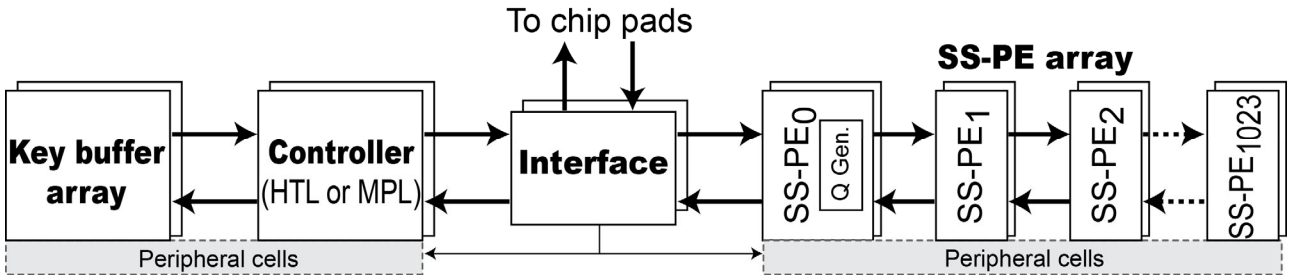
ゲートレベルハンドシェーク

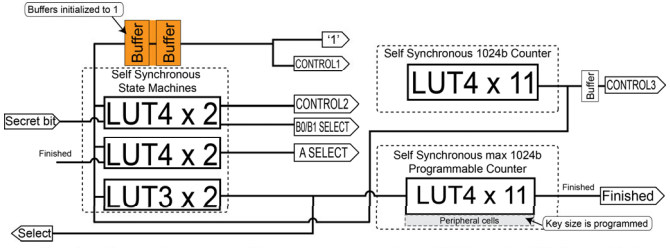
- ☺ **Precharge time is concealed**
  - **But guaranteed when $t_{precharge}<t_{evaluation}$**
- ☺ **Small CD delay**
- ☺ **No explicit latches**
- ☹ **Area overhead (~66%)**



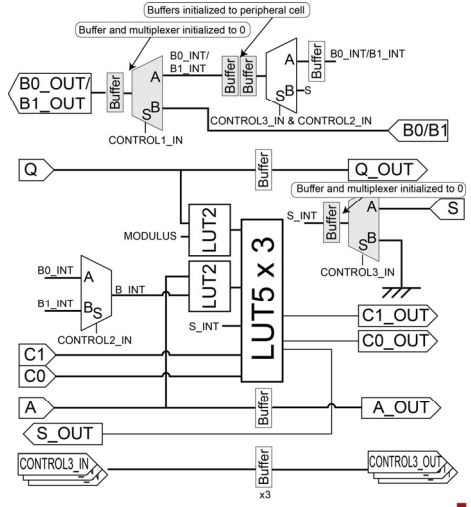


ゲートレベルハンドシェーク方式によるRSA



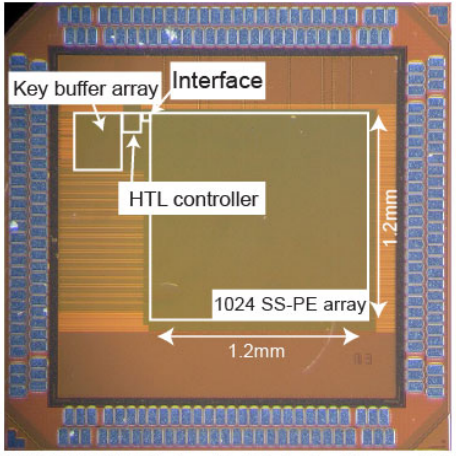


Controller (state machine programming differs for HTL & MPL)

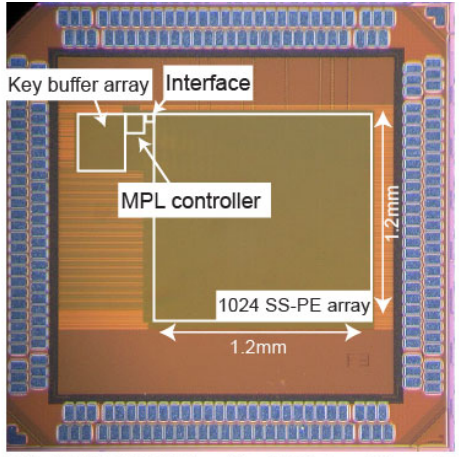


40nm CMOSによる実装

- 40nm CMOS
- 1024 bit RSA
- Two chips are fabricated, for HTL and MPL algorithm
- 201k gates used (M-SSRSA was 178k)



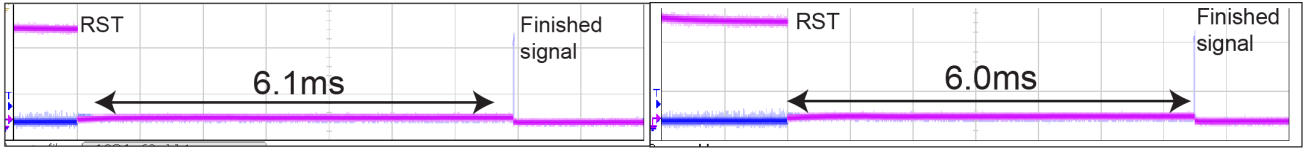
(a) 1024-bit SSRSA with HTL in 40nm CMOS



(b) 1024-bit SSRSA with MPL in 40nm CMOS

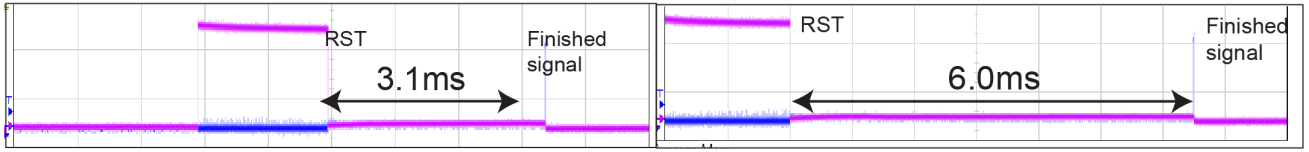
RSA実行結果と動作時の電流波形

- **Correct operation measured at 1.1V. HTL (a,b) shows key-dependent and MPL (c,d) key-independent operation time**
- **SPA current waveform show no information leaked (f,g)**



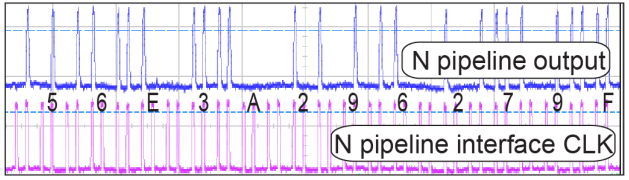
(a) HTL, Secret key all 1 (2 interleaved operations)

(c) MPL, Secret key all 1 (2 interleaved operations)

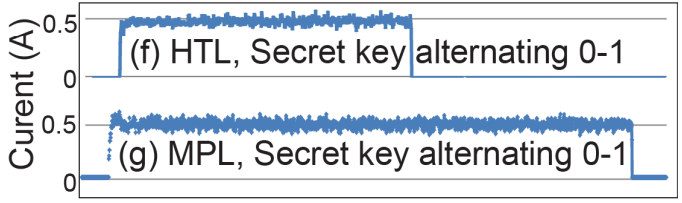


(b) HTL, Secret key all 0 (2 interleaved operations)

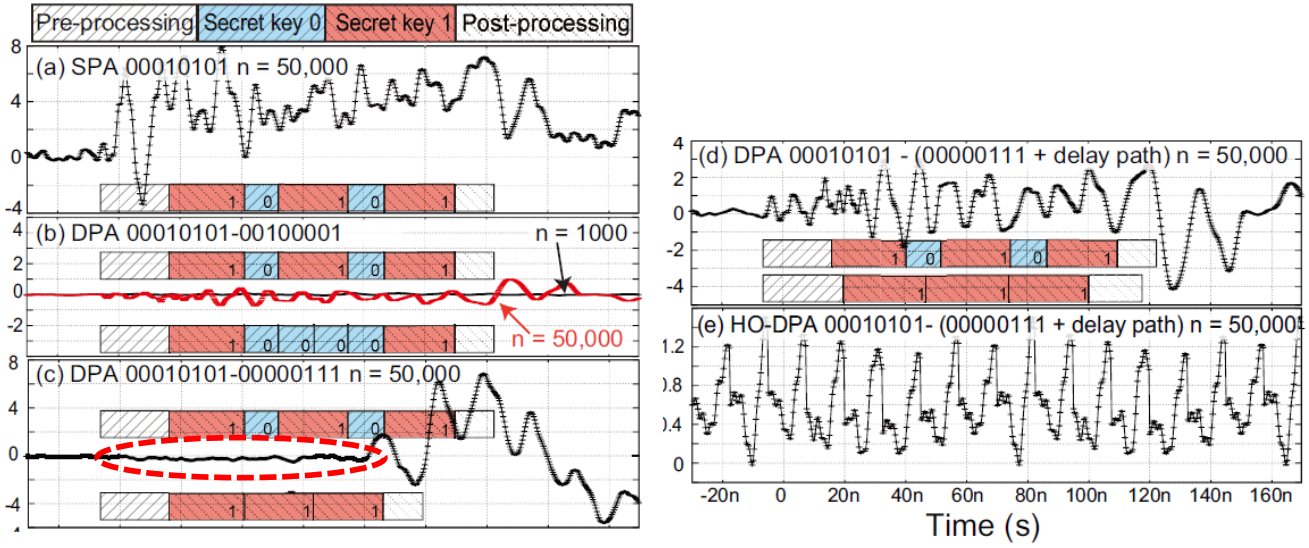
(d) MPL, Secret key all 0 (2 interleaved operations)



(e) Reading out correct value after operation from N pipeline



8ビットRSAの耐タンパー性評価



No key related information, except key length, is revealed
Key length issue can be solved using Montgomery Power Ladder Algorithm

内容

- 背景
- 楕円曲線暗号向けハードウエアの実現
- 高機能暗号と準同形暗号
- 暗号と安全性
- まとめ

まとめ

- ECDSA
  - 設計空間の中で、基数、座標系、アルゴリズムなどと面積・遅延時間などの見積もりを行い、目的にあった設計を選定できるようになった
- ペアリング
  - 高機能暗号の実用化に向けて高性能ペアリングエンジンの設計を目指し、65nmで30usでのペアリング演算を実現した
  - 先端プロセスを用いることでさらに高速化が可能である種の秘匿検索を1秒以内で実現する可能性・属性暗号の高速化に寄与
  - ペアリング曲線ごとのセキュリティ評価
- その先：耐量子計算暗号・準同形暗号
- 耐タンパ性
  - アルゴリズム的にはモンゴメリラダー法を利用することで耐性を持たせることが可能、ただし実装時に演算順、書き込みタイミングを精査する必要性がある