Field Instruction Multiple Data

Khin Mi Mi Aung [1], Enhui Lim [1], **Jun Jie Sim** [1 2], Benjamin Hong Meng Tan [1], Huaxiong Wang [2], Sze Ling Yeo

[1]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore

[2]School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore

2 June 2022

Homomorphic Encryption

Homomorphic Encryption (HE) [Gen09]

$$\begin{tikzcd}
\llbracket m \rrbracket \arrow[r, "f_{hom}"] & f(\llbracket m \rrbracket) = \llbracket f(m) \rrbracket \arrow[d, "\text{Decrypt}"] \\
m \arrow[u, "\text{Encrypt}"] \arrow[r, dashed, "f"'] & f(m)
\end{tikzcd}$$

Applications in

- Bioinformatics - analyzing sensitive genomic data
- Finance - KYC

SIMD Packing

Single Instruction, Multiple Data (SIMD) [SV11]

- Pack multiple data into a single ciphertext

$$\boxed{a}\boxed{b}\boxed{c}\boxed{d}.$$

- Applying the function f on the ciphertext is equivalent to simultaneously applying f on all data

$$f(\boxed{a}\boxed{b}\boxed{c}\boxed{d}) \longrightarrow \boxed{f(a)}\boxed{f(b)}\boxed{f(c)}\boxed{f(d)}.$$

- SIMD rotate/shift

$$\boxed{a}\boxed{b}\boxed{c}\boxed{d} \longrightarrow \boxed{c}\boxed{d}\boxed{a}\boxed{b}.$$

- Improve efficiency of HE schemes (BFV, BGV) by reducing the number of ciphertext needed.

Plaintext Space Decomposition

- Encode multiple data into a plaintext space R_t before encryption.
- Decomposition of the ring R_t into "slots" (Chinese Remainder Theorem)

$$R_t \simeq {}^{\mathbb{Z}[x]}\big/_{F_1(x)} \times \cdots \times {}^{\mathbb{Z}[x]}\big/_{F_\ell(x)} \mod t.$$

- Each slot (factor) is isomorphic to an finite extension field

$${}^{\mathbb{Z}_t[x]}\big/_{F_i(x)} \simeq \mathbb{F}_{t^d}.$$

Parameter Choice for Homomorphic Encryption

- Choose powers-of-2 cyclotomics of degree n for R_t
  - HE Standarization
  - Fast negacyclic FFT

Parameter Choice for Homomorphic Encryption

- Choose powers-of-2 cyclotomics of degree n for R_t
  - HE Standarization
  - Fast negacyclic FFT
- Maximize number of slots $\Rightarrow$ large prime t needed.

n	prime, t	max slots, ℓ	degree, d
	3	2	2048
	7	4	1024
4096	127	64	64
	12799	16	256
	40961	4096	1

Table: Choice of prime and number of slots

Parameter Choice for Homomorphic Encryption

- Choose powers-of-2 cyclotomics of degree n for R_t
  - HE Standarization
  - Fast negacyclic FFT
- Maximize number of slots $\Rightarrow$ large prime t needed.

n	prime, t	max slots, ℓ	degree, d
	3	2	2048
	7	4	1024
4096	127	64	64
	12799	16	256
	40961	4096	1

Table: Choice of prime and number of slots

- Can we use smaller prime and encode as much data?

A “Homomorphism” between a Vector Space and a Field

$(k, w)_t$-RMFE [Cas+18]

Embed a length k vector $\in (\mathbb{F}_t)^k$ into an finite extension field $\mathbb{F}_{t^w}$ of degree $w < d$ via Riemann Roch spaces $\mathcal{L}(G)$, $\mathcal{L}(2G)$.

$$\begin{array}{ccccc} (\mathbb{F}_t)^k & \rightleftharpoons & \mathcal{L}(G) \subseteq \mathcal{L}(2G) & \rightleftharpoons & \mathbb{F}_{t^w} \\ \text{Vector Space} & & \text{Riemann Roch Spaces} & & \text{Extension Field} \end{array}$$

- A Riemann Roch space is a special set of polynomials defined over $\mathbb{F}_q(\mathcal{C})$, the function field over a curve $\mathcal{C}$.
- Additive and multiplicative “homomorphism” between $(\mathbb{F}_t)^k$ and $\mathbb{F}_{t^w}$
  - $(x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y_2) \simeq \mathcal{X} + \mathcal{Y}$
  - $(x_1, x_2) \times (y_1, y_2) = (x_1 y_1, x_2 y_2) \simeq \mathcal{X} \cdot \mathcal{Y}$

RMFE Maps

RMFE Maps [Cas+18]

$$(\mathbb{F}_t)^k \underset{\pi}{\rightleftharpoons} \mathcal{L}(G) \subseteq \mathcal{L}(2G) \overset{\tau}{\rightleftharpoons} \mathbb{F}_{t^w}$$

Vector Space | Riemann Roch Spaces | Extension Field

- Encode, $\phi = \tau \circ \pi^{-1} : (\mathbb{F}_t)^k \to \mathbb{F}_{t^w}$
- Decode, $\psi = \pi \circ \tau^{-1} : \mathbb{F}_{t^w} \to (\mathbb{F}_t)^k$

- π and τ are linear maps $\Rightarrow$ ϕ and ψ are linear maps.

RMFE Maps

RMFE Maps [Cas+18]

$$\begin{tikzcd} (\mathbb{F}_t)^k \arrow[r, shift left] & \mathcal{L}(G) \subseteq \mathcal{L}(2G) \arrow[l, shift left, "\pi"] \arrow[r, shift left, "\tau"] & \mathbb{F}_{t^w} \arrow[l, shift left] \end{tikzcd}$$

Vector Space | Riemann Roch Spaces | Extension Field

- Encode, $\phi = \tau \circ \pi^{-1} : (\mathbb{F}_t)^k \to \mathbb{F}_{t^w}$
- Decode, $\psi = \pi \circ \tau^{-1} : \mathbb{F}_{t^w} \to (\mathbb{F}_t)^k$
- Recode, $\phi \circ \psi : \mathbb{F}_{t^w} \to \mathbb{F}_{t^w}$

- π and τ are linear maps $\Rightarrow$ ϕ and ψ are linear maps.
- To support multiplication, τ is defined from $\mathcal{L}(2G)$.
  - If $f, g \in \mathcal{L}(G)$, then $fg \in \mathcal{L}(2G)$
  - Decode correctly after ≤ 1 multiplication.

Field Instruction Multiple Data (FIMD)

Field Instruction Multiple Data (FIMD)

- $\mu \in R_t \leftarrow$ $\text{FIMD.Encode}\left(\boxed{x_1 \mid \cdots \mid x_k}, \ldots, \boxed{x_{k(\ell-1)+1} \mid \cdots \mid x_{k\cdot\ell}}\right)$
  1. $\boxed{\hat{x}_i} \leftarrow \text{RMFE.Encode}\left(\boxed{x_{k(i-1)+1} \mid \cdots \mid x_{k\cdot i}}\right)$ for $i = 1, \ldots, \ell$
  2. $\mu \in \mathbb{F}_{t^d} \leftarrow \text{SIMD.Encode}\left(\boxed{\hat{x}_1}, \ldots, \boxed{\hat{x}_\ell}\right)$

- $\mathbf{m} \in (\mathbb{F}_t)^{k\cdot\ell} \leftarrow \text{FIMD.Decode}(\mu \in R_t)$
  1. $\boxed{\hat{m}_1 \mid \cdots \mid \hat{m}_\ell} \in (\mathbb{F}_t)^\ell \leftarrow \text{SIMD.Decode}(\mu)$
  2. $\mathbf{m} \in (\mathbb{F}_t)^{k\cdot\ell} \leftarrow \left(\text{RMFE.Decode}\left(\boxed{\hat{m}_1}\right), \ldots, \text{RMFE.Decode}\left(\boxed{\hat{m}_\ell}\right)\right)$

FIMD Operations

FIMD Operations

- $c^{+} \leftarrow \text{FIMD.Add}(\hat{\mu}_1, \hat{\mu}_2)$

$$c^{+} = \text{HE.Add}(\hat{\mu}_1, \hat{\mu}_2).$$

- $c^{\times} \leftarrow \text{FIMD.Mult}(\hat{\mu}_1, \hat{\mu}_2)$

$$c^{\times} = \text{RMFE.Recode}(\text{HE.Mult}(\hat{\mu}_1, \hat{\mu}_2)).$$

- $c^{'} \leftarrow \text{FIMD.Rotate/Shift}(c, p)$.
  1. Write $p = p_{\text{SIMD}} k + p_{\text{RMFE}}$
  2. If $p_{\text{RMFE}} = 0$, $c^{'} = \text{HE.Rotate/Shift}(c, p)$.
  3. Else, $c^{'} = \text{HE.Rotate/Shift}(\text{RMFE.Rotate/Shift}(c, p_{\text{RMFE}}))$

RMFE Extensions

Two RMFE extensions for FIMD

- r-fold RMFE
- 3-Stage-Recode for Composite RMFE

r-fold RMFE

r-fold RMFE

Define τ such that r multiplications can be supported before recoding.

$$\tau : \mathcal{L}(2^r G) \longrightarrow \mathbb{F}_{t^w}$$

Assign a tag η to each FIMD ciphertext such that if $\eta = r$, recode.

- Fully utilize the whole field extension if $\dim(\mathcal{L}(2G)) \ll d$.
- Reduce the number of RMFE.Recodes
- Interoperability between data multiplied different number of times, e.g. $a \cdot b$ for $a \in \mathcal{L}(2G)$ and $b \in \mathcal{L}(4G)$.

Composite RMFE

Composite RMFE [Cas+18]

- Inner $(k_{\text{in}}, w_{\text{in}})_t$-RMFE
  - $\phi_{\text{in}} : (\mathbb{F}_t)^{k_{\text{in}}} \to \mathbb{F}_{t^{w_{\text{in}}}}$
  - $\psi_{\text{in}} : \mathbb{F}_{t^{w_{\text{in}}}} \to (\mathbb{F}_t)^{k_{\text{in}}}$
- Outer $(k_{\text{out}}, w_{\text{out}})_{t^{w_{\text{in}}}}$-RMFE
  - $\phi_{\text{out}} : (\mathbb{F}_{t^{w_{\text{in}}}})^{k_{\text{out}}} \to \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}}$
  - $\psi_{\text{out}} : \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}} \to (\mathbb{F}_{t^{w_{\text{in}}}})^{k_{\text{out}}}$

Use smaller RMFEs to build a $(k_{\text{in}} k_{\text{out}}, w_{\text{in}} w_{\text{out}})_t$-RMFE

$$\phi : (\mathbb{F}_t)^{k_{\text{in}} k_{\text{out}}} \to \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}} \qquad \psi : \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}} \to (\mathbb{F}_t)^{k_{\text{in}}}$$

$$\phi(\cdot) = \phi_{\text{out}}\left(\phi_{\text{in}}\left(\boxed{x_{1,1} \mid \cdots \mid x_{1,k_{\text{in}}}}\right), \cdots, \phi_{\text{in}}\left(\boxed{x_{k_{\text{out}},1} \mid \cdots \mid x_{k_{\text{out}},k_{\text{in}}}}\right)\right)$$

$$\psi(\cdot) = \left(\psi_{\text{in}}\left(\boxed{\psi_{\text{out}}[1]}\right), \cdots, \psi_{\text{in}}\left(\boxed{\psi_{\text{out}}[k_{\text{out}}]}\right)\right)$$

Composite RMFE

$(k_{\text{in}} k_{\text{out}}, w_{\text{in}} w_{\text{out}})_t$ Composite RMFE

Let (ϕ, ψ) be a $(k_{\text{in}} k_{\text{out}}, w_{\text{in}} w_{\text{out}})_t$ composite RMFE

$$\phi : (\mathbb{F}_t)^{k_{\text{in}} k_{\text{out}}} \to \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}}$$

$$\psi : \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}} \to (\mathbb{F}_t)^{k_{\text{in}} k_{\text{out}}}$$

- Decompose $\mathbb{F}_{t^w}$ into a tower of field extensions of $\mathbb{F}_t$ with $w_{\text{in}} w_{\text{out}} \le w$

$$\mathbb{F}_t \subseteq \mathbb{F}_{t^{w_{\text{in}}}} \subseteq \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}} .$$

- Reduce the cost of RMFEs used (size of linear maps).
  - E.g. Mapping between $(\mathbb{F}_3)^k$ and $\mathbb{F}_{3^{2048}}$, with intermediate field $\mathbb{F}_{3^{16}}$:
  - Direct RMFE encode: $k_{\text{in}} k_{\text{out}} \times 2048$ matrix.
  - Composite RMFE encode: k_{out} many inner $k_{\text{in}} \times 16$ matrices and outer $k_{\text{out}} \times 128$ matrix.

3-Stage-Recode - Composite RMFE

3-Stage-Recode

$$(\mathbb{F}_t)^{k_{\text{in}} k_{\text{out}}} \underset{2b.\ k_{\text{out}}\ \text{many}\ \phi_{\text{in}}}{\overset{2a.\ k_{\text{out}}\ \text{many}\ \psi_{\text{in}}}{\leftrightarrows}} (\mathbb{F}_{t^{w_{\text{in}}}})^{k_{\text{out}}} \underset{3.\ \phi_{\text{out}}}{\overset{1.\ \psi_{\text{out}}}{\leftrightarrows}} \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}}$$

- Step 1 and 3 work over the intermediate field $\mathbb{F}_{t^{w_{\text{in}}}}$.
- Step 2 work over $\mathbb{F}_t$.

3-Stage-Recode - Composite RMFE

3-Stage-Recode

$$(\mathbb{F}_t)^{k_{\text{in}} k_{\text{out}}} \underset{2b.\ k_{\text{out}} \text{ many } \phi_{\text{in}}}{\overset{2a.\ k_{\text{out}} \text{ many } \psi_{\text{in}}}{\rightleftarrows}} (\mathbb{F}_{t^{w_{\text{in}}}})^{k_{\text{out}}} \underset{3.\ \phi_{\text{out}}}{\overset{1.\ \psi_{\text{out}}}{\rightleftarrows}} \mathbb{F}_{t^{w_{\text{in}} w_{\text{out}}}}$$

- Step 1 and 3 work over the intermediate field $\mathbb{F}_{t^{w_{\text{in}}}}$.
- Step 2 work over $\mathbb{F}_t$.
- Optimization: Extend to r-fold RMFE for both inner and outer RMFEs.
- Delay inner recode (Step $2a, 2b$) until necessary.

RMFE Parameters

Curve $\mathcal{C}$	Base Field	Max k	Genus g	w
Projective	$\mathbb{F}_t$	$t+1$	0	
Elliptic	$\mathbb{F}_t$	$t+1+2\sqrt{t}$	1	$2k+4g-1$
Hermitian	${}^{\star}\mathbb{F}_{t^2}$	t^3+1	$\frac{t(t-1)}{2}$	

Table: Possible RMFE parameters with particular curves

⋆ With Hermitian curves, the slot degree d is effectively halved.

Experimental Setup

- Choice of primes $t = 3, 7$.
- HE parameters set to 80-bit security
  - $n = 4096$
  - $\max \log p = 159$
- Repeated squaring of one FIMD ciphertext and one HE ciphertext until decryption fails.
- Record the time and number of multiplications.
- Amortized speedup between FIMD ciphertext multiplication and HE ciphertext multiplication.

1. Homomorphic Encryption
2. Reverse Multiplication Friendly Embeddings (RMFE)
3. This Work
   - r-fold RMFEs
   - 3-Stage-Recode for Composite RMFEs
4. Experimental Results
   - r-fold RMFE Results
   - Composite RMFE Results
5. Conclusion
6. References

r-fold RMFE Results

t-Field	k	r	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
7-P	8	1	$3:5$	3.7435	$0.0516\times$
		4	$4:5$	1.1595	$0.2606\times$
		$4^{\star}$	$4:5$	0.0431	$7.4935\times$
7-E	13	1	$3:5$	3.7146	$0.0853\times$
		4	$4:5$	0.9307	$0.5362\times$
		$4^{\star}$	$4:5$	0.0478	$10.9844\times$
7-H	214	1	$3:5$	1.8301	$2.7884\times$

*No recodes were performed

Table: (r)FIMD Mult Noise Consumption

r-fold RMFE Results

t-Field	k	r	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
7-P	8	1	3 : 5	3.7435	0.0516×
		4	4 : 5	1.1595	0.2606×
		$4^\star$	4 : 5	0.0431	7.4935×
7-E	13	1	3 : 5	3.7146	0.0853×
		4	4 : 5	0.9307	0.5362×
		$4^\star$	4 : 5	0.0478	10.9844×
7-H	214	1	3 : 5	1.8301	2.7884×

*No recodes were performed

Table: Better amortized time/speedup as r increases

r-fold RMFE Results

t-Field	k	r	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
7-P	8	1	3 : 5	3.7435	0.0516×
		4	4 : 5	1.1595	0.2606×
		4*	4 : 5	0.0431	$\boxed{7.4935\times}$
7-E	13	1	3 : 5	3.7146	0.0853×
		4	4 : 5	0.9307	0.5362×
		4*	4 : 5	0.0478	$\boxed{10.9844\times}$
7-H	214	1	3 : 5	1.8301	$\boxed{2.7884\times}$

*No recodes were performed

Table: Higher k gives better amortized speedup

Composite RMFE Results

t-Field	k	r	$^{d}\!/_{k}$	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
3-E	7	1	293	$\boxed{3:6}$	7.16	$0.0176\times$

t-Field	$(k_{\text{in}}, r_{\text{in}})_t$	$(k_{\text{out}}, r_{\text{out}})_{t^{d'}}$	$^{d}\!/_{k_{\text{total}}}$	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
C3-E	$(6,1)_3$	$(64,1)_{3^{16}}$	5.33	$\boxed{2:6}$	2.434	$1.9544\times$

Table: Noise Consumption in Composite RMFE vs r-fold RMFE

Composite RMFE Results

t-Field	k	r	$^{d}\!/_{k}$	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
3-E	7	1	$\boxed{293}$	3 : 6	7.16	$\boxed{0.0176\times}$

t-Field	$(k_{\text{in}}, r_{\text{in}})_t$	$(k_{\text{out}}, r_{\text{out}})_{t^{d'}}$	$^{d}\!/_{k_{\text{total}}}$	(r)FIMD:HE Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
C3-E	$(6,1)_3$	$(64,1)_{3^{16}}$	$\boxed{5.33}$	2 : 6	2.434	$\boxed{1.9544\times}$

Table: Packing Improvements in Composite RMFE vs r-fold RMFE

Composite RMFE Results

t-Field	k	w	r	Max (r)FIMD Mult	(r)FIMD Mult (sec)	1 (r)FIMD Mult (sec)
C7-P	$8 \cdot 32 = 256$	$16 \cdot 63 = 1008$	$(1, 1)$	1	0.754	0.754
7-H	214	511	1	3	1.83	0.610

Table: 3-Stage-Recode versus Direct Recode

- Did not compute direct recode map for C7-P (too large).
- Theoretical w_{total} for C7-P is $16 \cdot 63 = 1008$.
- Extrapolated direct recode time C7-P based on 7-H

$$0.610 \cdot \frac{1008}{511} = 1.203 \text{ seconds.}$$

Composite RMFE Results

t-Field	$(k_{\text{in}}, r_{\text{in}})_t$	$(k_{\text{out}}, r_{\text{out}})_{t^{d'}}$	k_{total}	(r)FIMD Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
C3-E	$(2,2)_3$	$(64,1)_{3^{16}}$	128	2	1.4284200	1.138740×
	$(6,1)_3$	$(64,1)_{3^{16}}$	384	2	2.4344100	1.954400×
C3-H	$(3,2)_3$	$(16,1)_{3^{64}}$	48	2	0.4838210	1.189630×
	$(11,1)_3$	$(16,1)_{3^{64}}$	176	2	1.1126300	1.891190×
	$(11,2)_3$	$(8,1)_{3^{128}}$	88	2	0.4557780	2.271160×
	$(27,1)_3$	$(8,1)_{3^{128}}$	216	2	1.0570600	2.641760×

Table: Effect of varying r_{in}, keeping r_{out} and d' fixed

Composite RMFE Results

t-Field	$(k_{\text{in}}, r_{\text{in}})_t$	$(k_{\text{out}}, r_{\text{out}})_{t^{d'}}$	k_{total}	(r)FIMD Mult	(r)FIMD Mult Time (sec)	Amortized Speedup
C3-E	$(2,2)_3$	$(64,1)_{3^{16}}$	128	2	1.4284200	1.138740×
	$(6,1)_3$	$(64,1)_{3^{16}}$	384	2	2.4344100	1.954400×
C3-H	$(3,2)_3$	$(16,1)_{3^{64}}$	48	2	0.4838210	1.189630×
	$(11,1)_3$	$(16,1)_{3^{64}}$	176	2	1.1126300	1.891190×
	$(11,2)_3$	$(8,1)_{3^{128}}$	88	2	0.4557780	2.271160×
	$(27,1)_3$	$(8,1)_{3^{128}}$	216	2	1.0570600	2.641760×

Table: Effect of varying r_{in}, keeping r_{out} and d' fixed

- Balance between multiplication timing and amount of data to pack.

1. Homomorphic Encryption
2. Reverse Multiplication Friendly Embeddings (RMFE)
3. This Work
   - r-fold RMFEs
   - 3-Stage-Recode for Composite RMFEs
4. Experimental Results
   - r-fold RMFE Results
   - Composite RMFE Results
5. **Conclusion**
6. References

Conclusion

- Allow small primes with Homomorphic Encryption
  - Almost the same amount of packed data.
- Two RMFE extensions
  - r-fold RMFE
  - 3-Stage-Recode with Composite RMFE
- Tradeoff when using FIMD
  - FIMD multiplication consumes more noise.
  - Amortized speedup when using FIMD.
  - Some form of balancing between running time and amount of data to pack.

Thank you

1. Homomorphic Encryption

2. Reverse Multiplication Friendly Embeddings (RMFE)

3. This Work
   - r-fold RMFEs
   - 3-Stage-Recode for Composite RMFEs

4. Experimental Results
   - r-fold RMFE Results
   - Composite RMFE Results

5. Conclusion

6. References

References I

[Gen09] Craig Gentry. "Fully Homomorphic Encryption Using Ideal Lattices". In: STOC '09. 2009. URL: https://doi.org/10.1145/1536414.1536440.

[SV11] N.P. Smart and F. Vercauteren. *Fully Homomorphic SIMD Operations*. Cryptology ePrint Archive, Report 2011/133. https://ia.cr/2011/133. 2011.

[Cas+18] Ignacio Cascudo et al. *Amortized Complexity of Information-Theoretically Secure MPC Revisited*. Cryptology ePrint Archive, Report 2018/429. https://ia.cr/2018/429. 2018.

[FV12] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2012/144. https://ia.cr/2012/144. 2012.

References II

[BGV11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. *Fully Homomorphic Encryption without Bootstrapping*. Cryptology ePrint Archive, Report 2011/277. https://ia.cr/2011/277. 2011.

[Che+16] Jung Hee Cheon et al. *Homomorphic Encryption for Arithmetic of Approximate Numbers*. Cryptology ePrint Archive, Report 2016/421. https://ia.cr/2016/421. 2016.