А. А. Хлюпин, А. О. Саакян, О. В. Ниссенбаум
Тюменский государственный университет, г. Тюмень
УДК 004.056.55

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ШИФРОВАНИЯ ДЛЯ БЕЗОПАСНЫХ МНОГОСТОРОННИХ ВЫЧИСЛЕНИЙ

Аннотация. В этой работе были рассмотрены стандартизированные блочные шифры, а также алгоритмы шифрования (псевдослучайные функции), специально разработанные для работы в режиме многосторонних безопасных вычислений. Был рассмотрен перспективный алгоритм шифрования, созданный на его основе 2-сторонний протокол шифрования и выделена проблема расширения этого протокола на большее число участников.

Ключевые слова: MPC, алгоритм шифрования, протокол шифрования, псевдослучайная функция, препроцессинг, многосторонние протоколы, протоколы безопасных вычислений.

Введение. Многосторонние безопасные вычисления (MPC-Multiparty computation) — криптографические протоколы, позволяющие нескольким участникам вычислить некоторую функцию от своих конфиденциальных входных данных. Безопасность таких протоколов определяется в определенной модели противника, который может скомпрометировать некоторое подмножество участников протокола. Пассивный противник получает доступ на чтение ко всем данным скомпрометированных участников, но не способен вмешиваться в выполнение протокола. Активный противник, кроме того, способен заставить скомпрометированных участников произвольно отклоняться от протокола. Неформально, безопасность понимается как невозможность для злоумышленника получить секретные данные и повлиять на результат протокола иначе, чем изменением входных данных скомпрометированных участников.

В настоящее время для обеспечения конфиденциальности информации разработчики приложений и программного обеспечения все чаще используют протоколы безопасных многосторонних вычислений. Например, протоколы MPC успешно применяются корпорацией Google для обеспечения безопасности аутентификации пользователей [1], своей облачной инфраструктуры [2] или представленной недавно ими Private Join and Compute [3]. С каждым

годом появляется все больше поставщиков технологий MPC, например MPC альянс [4] насчитывает почти 60 вендоров. Большинство практически применимых протоколов MPC опираются на вычислительную стойкость таких примитивов как блочные шифры и псевдослучайные функции.

Например, протокол BMR [5] является протоколом MPC построенным на основе многосторонней реализации AES [7], но, из-за невозможности полной оптимизации AES для работы в распределенной среде, все его вычисления производятся локально из-за чего размер блока увеличивается в зависимости от увеличения числа участников. Для того чтобы избежать такого большого потребления памяти, а также добиться еще большей скорости вычислений, необходимо использовать такие протоколы MPC, которые были бы основаны на блочных шифрах или псевдослучайных функциях эффективно реализуемых для MPC. Например, в 2012 г. был представлен шифр LowMC [10], в котором авторы уменьшили число раундов до 40 при сравнимым с AES уровнем безопасности, то есть снизили раундовую сложность своего алгоритма. Позже был представлен новый шифр RASTA [11], в котором были представлены вариации шифра с максимальной раундовой сложностью всего в 6 раундов, но эффективными из них были только вариации с от 4 до 6 раундов. В 2021 г. был представлен алгоритм MPC-friendly, который, как показали авторы статьи, обладает всего лишь 2 раундами. Дальнейшее рассмотрение приведенных выше статей идет в следующих разделах.

Проблема исследования. В современных протоколах MPC широко используются блочные шифры. Как пример можно привести использование блочных шифров в протоколе BMR [5], где блочный шифр вызывается участниками каждый раз, когда им надо перемножить 2 секретных бита, а это миллионы раз. Современные блочные шифры (псевдослучайные функции), такие как AES [6], при выполнении их локально выполняются с приемлемой скоростью, но при распределенном вычислении шифра они показывают не удовлетворительную скорость работы. Например, в протоколе BMR каждый участник вычисляет блочный шифр локально со своим секретным ключом, что кратно увеличивает суммарную длину секретного

ключа. Один из главных подходов в многосторонних вычислениях предполагает хранение секретных данных участниками в виде разделенного секрета. В случае применения такого подхода к секретным ключам BMR, можно было бы сделать размер контура независимым от числа участников. Однако, такой подход требует алгоритмов шифрования, эффективно вычислимых в режиме MPC с разделенным ключом. Поэтому в последнее время появилось большое количество шифров, основанных на иных принципах, нежели традиционные SP-сети, многораундовая структура которых препятствует эффективной многосторонней реализации.

Настоящая работа направлена на анализ эффективности некоторых новых алгоритмов шифрования при использовании в многостороннем режиме. В данной работе произведен:

- анализ алгоритмов шифрования, с точки зрения их применимости в режиме MPC;
- подготовка информационной базы для дальнейшей многосторонней реализации одного из выбранных алгоритмов.

Материалы и методы. *Анализ эффективности многосторонних реализаций протокола AES.* AES — это стандартизированный симметричный блочный шифр, являющийся не только национальным стандартом США, но и стандартом де-факто в подавляющем количестве криптографических приложений. Это привело к созданию очень эффективных программных и аппаратных реализаций AES, которые позволяют шифровать миллионы блоков в секунду. Однако шифр не был разработан с учетом многосторонних вычислений. Хотя существует множество реальных применений MPC, требующих блочных шифров, применение стандартных шифров, таких как AES, не эффективно. Например, в работе [7] авторами было предложено использование обратного AES, а также его улучшение через интеграцию SPDZ [8] в препроцессинг и оптимизации этапа локальных вычислений. По сравнению с результатами, описанными в статье 2012 г. [9], где применялся прямой AES, в их работе потребление памяти упало больше чем в 2 раза, а время вычисления одного блока в 3 раза (табл. 1).

Таблица 1

Сравнение прямого и обратного AES

	Реализация [9]	*Реализация [8]*	*Коэффициент улучшения*
Хранение (Кбайт)	54.4	20.08	x 2.7
Передача (Кбайт)	18.27	8.2	x 2.23
Время выполнения (миллисекунды)	5.026	1.5	x 3.35

Основная проблема при распределенном вычислении заключается в количестве раундов передачи данных между сторонами на что непосредственно влияет число раундов самого шифра. Шифр AES имеет от 10 до 14 раундов подстановочно-перестановочной сети, каждый из которых требует, в свою очередь, нескольких раундов передачи данных в режиме MPC. Таким образом, эффективность алгоритма AES в режиме MPC почти достигла предела своей возможной оптимизации, то есть дальнейшие попытки оптимизации будут приводить только к улучшению эффективности на определенную константу.

Анализ существующих шифров (псевдослучайных функций) для MPC. В последнее время появилось большое количество работ, посвященных разработке шифров (псевдослучайных функций), которые были бы эффективно реализуемы в режиме многосторонних вычислений: LowMC [10], Rasta [11], MPC-friendly [12]. В работе [10] был предложен новый шифр, насчитывающий от 7 раундов коммуникаций. Для 128-битных ключей он требует 14 раундов, а при достижении такого же уровня безопасности, как AES, требует 40 раундов коммуникации. Авторами статьи [11] был предложен шифр RASTA, где были реализованы варианты с числом раундов коммуникации от 2 до 6, но они также не рекомендуют использовать 2-х и 3-х раундовые варианты, потому что они не обладают достаточными диффузионными свойствами, что негативно сказывается на безопасности секрета. К тому же, рекомендуемый размер блока для менее, чем 4-раундовых вариантов слишком велик (табл. 2).

Таблица 2

Минимальные размеры блоков от 2–6 раундов Rasta с целью обеспечения 80, 128, 256 бит безопасности

Уровень безопасности	*Раунды*				
	2	*3*	*4*	*5*	*6*
80 — бит	2320961	3939	327	327	219
128 — бит	9506325433	246831	1877	525	351
256 — бит	40829356287426654651	16167762975	445939	3545	703

В работе [12] была предложена псевдослучайная функция, в которой всего 2 раунда и отсутствуют проблемы, описанные в статье [11]. Так, для достижения 128-битовой безопасности достаточно использование 256-битных ключей, а все операции являются алгебраическими, такими как умножение и сложение матриц и векторов.

Анализ алгоритма MPC-friendly. MPC-friendly — алгоритм вычисления псевдослучайной функции, основанный на проблеме обучения с шумом (LPN — Learning-Parity-with-Noise) разработанный специально для работы в распределенном режиме. Стоит заметить, что это не первое использование LPN для создания алгоритмов MPC, например эту проблему можно встретить в этон научной работе [16]. По сравнению с другими шифрами (псевдослучайными функциями/генераторами) он включает в себя всего лишь два раунда. Авторами этой статьи были предложены следующие структурные элементы (рис. 1) для реализации этого примитива:

1. Lin (линейный) переход — получает на вход секретный x, умножает его на матрицу A и выводит результат $y = Ax$.
2. BL (билинейный) переход — получает на вход секретную матрицу K и секретный вектор x, выводит результат $y = Kx$.
3. Add переход — получает на вход x, x' и выводит результат $y = x + x' \, mod \, p$.
4. Преобразование $\mathbb{Z}_2 \rightarrow \mathbb{Z}_3$. Принимает в качестве входных данных вектор $x \in \mathbb{Z}_2$ и возвращает его эквивалентное представление в $x \in \mathbb{Z}_3$. В дальнейшем мы будем использовать упрощенную запись $Convert_{(2,3)}(x)$.

5. Преобразование $\mathbb{Z}_3 \to \mathbb{Z}_2$. Принимает в качестве входных данных вектор $x \in \mathbb{Z}_3$ и вычисляет его отображение в $x \in \mathbb{Z}_2$. В дальнейшем мы будем использовать упрощенную запись $Convert_{(3,2)}(x)$.

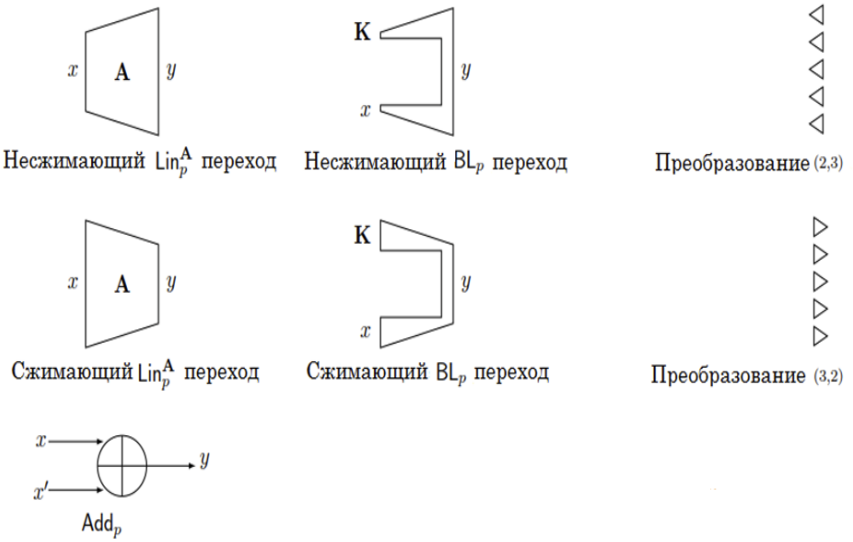

Рис. 1. Структурные элементы алгоритма MPC-friendly

Переходы Lin и Add вычисляются локально на общих входных данных и не требуют предобработки и коммуникаций, в то время как остальные переходы вычисляются в распределенной среде и требуют препроцессинга. Для лучшего понимания работы алгоритма MPC-friendly кратко рассмотрим функциональность протоколов преобразования $Convert_{(2,3)}(x)$ и $Convert_{(3,2)}(x)$ и их препроцессинг, а также реализацию препроцессинга в 2-стороннем протоколе предложенном авторами статьи.

Протоколы преобразования. Целью преобразования $Convert_{(2,3)}(x)$ является преобразование долей распределенного $x \in \mathbb{Z}_2$ в доли такого же $x^* = x \in \mathbb{Z}_3$. Этот протокол требует пре-

процессинга, который заключается в предоставлении каждой из сторон долей случайного бита $r^* = r$ над полями $\mathbb{Z}_2$ и $\mathbb{Z}_3$ как коррелированную случайность. Доли r над $\mathbb{Z}_2$ обозначим как $[\![r]\!] = \{r_1, \ldots, r_n\}$, а над $\mathbb{Z}_3$ как $[\![r^*]\!] = \{r_1^*, \ldots, r_n^*\}$, где n — количество участников MPC протокола, каждый из которых обладает соответствующей долей. К сожалению, авторам удалось получить эффективный протокол препроцессинга только для случая двух участников. Опишем его ниже.

Протокол препроцессинга $Convert_{(2,3)}(x)$.

1. Стороны P_1 и P_2 выполняют протокол забывчивой передачи, где P_1 выполняет роль отправителя со входом (z_0, z_1), где $z_0 \neq z_1$ — случайные числа из $\mathbb{Z}_3$, а P_2 — роль получателя со входом $c \in \mathbb{Z}_2$.
2. P_1 вычисляет пару (r_1, r_1^*), так что если $z_1 = z_0 - 1 \; mod \; 3$, то $(r_1, r_1^*) = (0, z_0)$, иначе если $z_0 = z_1 - 1 \; mod \; 3$, то $(r_1, r_1^*) = (1, z_1)$.
3. Сторона P_2 вычисляет $(r_2, r_2^*) = (c, -z_c \; mod \; 3)$.

К сожалению, такой протокол не расширяются на случай более чем двух участников, так как протокол забывчивой передачи (OT — Oblivious Transfer) [13] является двухсторонним.

Целью преобразования $Convert_{(3,2)}(x)$ является — преобразование долей распределенного $x \in \mathbb{Z}_3$ в доли такого же $x^* = x \in \mathbb{Z}_2$. Этот протокол требует препроцессинга, который заключается в предоставлении сторонам долей (по полю $\mathbb{Z}_2$) двух векторов, такие как: $u = u^* \; mod \; 2$ и $v = (v^* + 1 \; mod \; 3) \; mod \; 2$ как коррелированную случайность. Доли u и v над $\mathbb{Z}_3$ обозначим как $[\![u]\!] = \{u_1, \ldots, u_n\}$ и $[\![v]\!] = \{v_1, \ldots, v_n\}$, а над $\mathbb{Z}_2$ как $[\![u^*]\!] = \{u_1^*, \ldots, u_n^*\}$ и $[\![v^*]\!] = \{v_1^*, \ldots, v_n^*\}$, где n как и раньше — количество участников. Для преобразования $Convert_{(3,2)}(x)$ препроцессинг осуществляется аналогично преобразованию $Convert_{(2,3)}(x)$.

Отсутствие эффективного протокола препроцессинга для более чем 2-х участников препятствует применению данного алгоритма в многостороннем режиме. Тем не менее, существуют стандартные техники вычисления любой многосторонней функциональности, такие как GMW [14, 15]. В ходе дальнейшей работы мы планируем исследовать и применить эти стандартные техники для построения

препроцессинга, оценить их эффективность и сравнить получившуюся в результате реализацию MPC-friendly шифра с многосторонними реализациями традиционных блочных шифров, таких как AES.

Результаты. Нами был проведен анализ новых шифров с точки зрения их реализации в протоколах MPC выявлены их особенности и проблемы при многосторонней реализации.

Был произведен анализ работы алгоритма MPC-friendly, обозначены проблемы, затрудняющие его многостороннюю реализацию без доверенной стороны.

Заключение. В ходе работы нами была поставлена задача создания протокола препроцессинга для 3-х и более участников, используя стандартные техники, основанные на вычислительных контурах, а также реализации протокола шифрования MPC-friendly для 3-х и более участников, провести эксперименты для оценки его эффективности.

СПИСОК ЛИТЕРАТУРЫ

1. How Confidential Space and MPC can help secure digital assets | Google Cloud Blog. — Text: electronic // Google Cloud: [site]. — URL: https://cloud.google.com/blog/products/identity-security/how-confidential-space-and-mpc-can-help-secure-digital-assets (date of the application: 23.05.2023).
2. How to secure digital assets with multi-party computation and Confidential Space | Google Cloud Blog. — Text: electronic // Google Cloud: [site]. — URL: https://cloud.google.com/blog/products/identity-security/how-to-secure-digital-assets-with-multi-party-computation-and-confidential-space (date of the application: 23.05.2023).
3. Google Online Security Blog: Helping organizations do more without collecting more data. — Text: electronic // Google Security Blog: [site]. — URL: https://security.googleblog.com/2019/06/helping-organizations-do-more-without-collecting-more-data.html (date of the application: 24.05.2023).
4. MPC Alliance. — Text: electronic // MPC Alliance: [site]. — URL: https://mpcalliance.org (date of the application: 25.05.2023).
5. Beaver, D. The round complexity of secure protocols / D. Beaver, S. Micali, P. Rogaway. — Text: direct // Proceedings of the twenty-second annual ACM symposium on Theory of computing. — 1990. — P. 503-513.

6. Announcing the advanced encryption standard. — Text: electronic // Nation Institute of Standards and Technology: [site]. — URL: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf (date of the application: 24.05.2023).
7. Durak, F. B. Improving the Efficiency of AES Protocols in Multi-Party Computation / F. B. Durak, J. Guajardo. — Text: direct // Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part I 25. — 2021. — P. 229-248.
8. Multiparty computation from somewhat homomorphic encryption / I. Damgård, V. Pastro, N. Smart, S. Zakarias. — Text: direct // Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. — 2012. — P. 643-662.
9. Implementing AES via an Actively/Covertly Secure Dishonest-Majority MPC Protocol / I. Damgård, M. Keller, E. Larraia [et al.]. — Text: direct // Security and Cryptography for Networks: 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings 8. — 2012. — P. 241-263.
10. Ciphers for MPC and FHE / M. R. Albrecht, C. Rechberger, T. Schneider [et al.]. — Text: direct // Advances in Cryptology--EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I 34. — 2015. — P. 430-454.
11. Rasta: a cipher with low ANDdepth and few ANDs per bit / C. Dobraunig, M. Eichlseder, L. Grassi [et al.]. — Text: direct // Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part I 38. — 2018. — P. 662-692.
12. Mpc-friendly symmetric cryptography from alternating moduli: candidates, protocols, and applications / I. Dinur, S. Goldfeder, T. Halevi [et al.]. — Text: direct // Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part IV 41. — 2021. — P. 517-547.
13. Efficient Two-Round OT Extension and Silent Non-Interactive Secure Computation / E. Boyle, G. Couteau, N. Gilboa [et al.]. — Text: direct // Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. — 2019. — P. 291-308.

14. Micali, S. How to play any mental game / S. Micali, O. Goldreich, A. Wigderson. — Text: direct // Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC. — 1987. — P. 218-229.
15. Efficient Pseudorandom Correlation Generators: Silent OT Extension and More / E. Boyle, G. Couteau, N. Gilboa [et al.]. — Text: direct // Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39. — 219.— P. 489-518.
16. Exploring Crypto Dark Matter: New Simple PRF Candidates and Their Applications / D. Boneh, Y. Ishai, A. Passelègue [et al.]. — Text: direct // Theory of Cryptography: 16th International Conference, TCC 2018, Panaji, India, November 11–14, 2018, Proceedings, Part II. — 2018. — P. 699-729.