СПИСОК ЛИТЕРАТУРЫ:

1. *Gu Y., McCallum A., Towsley D.* Detecting anomalies in network traffic using maximum entropy // Tech. rep. Department of Computer Science. UMASS. Amherst, 2005.
2. *Lee W., Xiang D.* Information-theoretic measures for anomaly detection // Proceedings of the IEEE Symposium on Security and Privacy. IEEE Computer Society. 2001.
3. *Zhang H.* Study on the TOPN Abnormal Detection Based on the NetFlow Data Set // Computer and Information Science. 2009. Vol. 2. № 3.
4. Distributed Change-Point Detection of DDoS Attacks: Experimental Results on DETER Testbed // USENIX Association Berkeley. CA, USA.
5. *Maselli G., Deri L.* Design and Implementation of an Anomaly Detection System: an Empirical Approach // Terena TNC. Zagreb, Croatia, 2003.
6. *Sekar V., Duffield N., Spatscheck O., Van der Merwe J., Zhang H.* Lads: Large-scale automated DDoS detection system // USENIX Annual Technical Conference. 2006.
7. Cisco-Arbor Clean Pipe Solution 2.0 White Paper. URL: http://www.arbornetworks.com/index.php?option=com_docman&task=doc_download&gid=448.
8. Cisco Systems NetFlow Services Export Version 9 / Claise B., Ed. // RFC 3954. October 2004.
9. *Leinen S.* Evaluation of Candidate Protocols for IP Flow Information // RFC 3955. 2004.

А. Р. Орлов, Е. Б. Маховенко

АНАЛИЗ КРИПТОСИСТЕМ С ОТКРЫТЫМ КЛЮЧОМ, ПОСТРОЕННЫХ НА ОСНОВЕ ЗАДАЧ ТЕОРИИ РЕШЕТОК

Для построения криптосистем обычно применяются следующие задачи теории решеток:

1. По базису решетки найти кратчайший ненулевой вектор (SVP).

2. По базису решетки найти ближайший вектор $\bar{b}$ к заданному вектору $\bar{j}$ (CVP).

3. Найти m линейно независимых векторов B_m, для которых $\max_i \|Bx_i\| \le \lambda_n$ ($SIVP$).

4. Поиск кратчайшего расстояния между векторами в базисе решетки; по заданному базису определить, не превосходит ли кратчайший вектор норму N ($GapSVP$).

5. По базису решетки и заданному $\varepsilon > 0$ определить, существует ли в решетке вектор $\bar{v}$, близкий к заданному вектору $\bar{j}$ с точностью до ε ($GapCVP$).

6. По базису решетки, в котором кратчайший вектор в k раз меньше другого кратчайшего линейно независимого вектора, найти кратчайший вектор ($uSVP$).

7. Дана решетка с минимальным расстоянием l (необязательно известным); по базису B и вектору $\bar{j}$, такому, что $\rho(B, \bar{j}) \le \gamma l$, найти вектор в решетке (BDD).

8. Дан базис B q-нарной (модулярной) m-мерной решетки $L_q^{m \times n}$. Принадлежность вектора к решетке L определяется: $L(B) = \{B^T s \bmod q \subseteq Z^m, s \in Z^n\}$. На решетке равномерно распределен шум e (обычно с математическим ожиданием, равным 0, и дисперсией $\sqrt{q}$), q задан некоторым многочленом. $\bar{s} \in Z_q^n$ — некоторый исходный вектор без шума. Найти исходную точку в решетке (исключить шум) по некоторому множеству известных $(B\bar{s}_i + \bar{e}_i)$ — задача обучения с ошибками (LWE).

Все исследуемые криптосистемы можно условно разделить на два типа: имеющие строго доказанную криптостойкость, но неэффективные по времени работы и/или характеризующиеся быстрым ростом размеров ключей от параметров шифрования. К таким относятся системы на основе SVP, $uSVP$, $SIVP$ - задач [1]; эффективные по времени работы и затратам на хранение ключей, но не обладающие строго доказанной криптостойкостью. К таким относятся системы, основанные на частных случаях задач теории решеток или на решетках с цикличностью образующего их базиса [2], например криптосистема $NTRU$ (Draft standard IEEE 1363.1) [3].

Таким образом, необходимость устранения существующего противоречия между «теоретически криптостойкими» и «эффективными эмпирическими» решеточными системами асимметричного шифрования открывает перспективное направление фундаментальных и прикладных математических исследований в области постквантовой криптографии. Как же построить криптосистему, отвечающую всем предъявленным требованиям? Для решения данной проблемы необходимо выбрать систему так, чтобы для нее было возможно строго доказать криптостойкость и использовать параметры, соответствующие эффективному алгоритму шифрования, не забывая о проблеме роста размеров открытого/закрытого ключей с увеличением размерности базиса решетки.

Важным этапом в разработке криптосистемы является выбор задачи, на которой основывается ее криптостойкость. Среди рассматриваемых задач SVP-задача является самой сложной (NP-полной).

Криптосистема должна быть не только стойкой, но еще и эффективной по времени работы. Последним свойством обладают системы, основанные на частных случаях задач теории решеток или же основанные на решетках с цикличностью базиса. Но основная задача уже выбрана, поэтому возникает вопрос о совместимости задач. Составленный граф, отражающий возможность приведения задач теории решеток, представлен на рисунке 1.

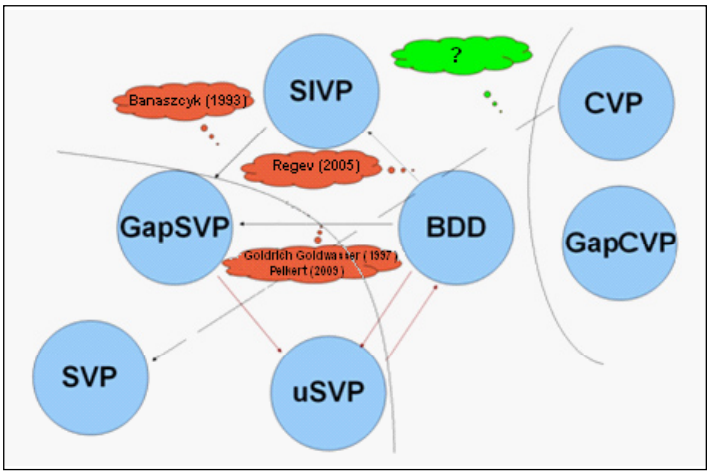

Рис. 1.

Полученные в ходе исследования рекомендации выглядят следующим образом:

— основополагающим моментом в построении криптосистемы должна стать самая трудная из всех задач теории решеток — SVP-задача, поскольку построенные таким образом криптосистемы будут иметь формальное доказательство криптостойкости;

— для достижения эффективности криптосистемы необходимо использовать иную задачу в совокупности с уже выбранной SVP-задачей, поскольку она не обеспечивает это условие;

— приоритетным является использование следующих сочетаний задач: $CVP \to SVP$ и $LWE \to SVP$, поскольку CVP- и LWE-задачи обеспечивают эффективные алгоритмы, поэтому параметры следует выбирать, руководствуясь принципами данных задач;

— строить криптосистему возможно на решетках с цикличностью образующего их базиса, это свойство способствует увеличению скорости работы системы.

Приведенные рекомендации теоретически позволяют построить криптосистему с доказуемой криптостойкостью и обеспечить эффективность ее работы, а также допустимые размеры ключей.

СПИСОК ЛИТЕРАТУРЫ:

1. *Silverman J. H.* An Introduction to the Theory of Lattices and Applications to Cryptography. 2006. — 76 p.
2. *Peikert C., Rosen A.* Efficient Collision-Resistant Hashing from worst-Case Assumptions on Cyclic Lattices. 2006. — 20 p.
3. *Hoffstein J., Pipher J., Silverman J. H.* NTRU: A ring-based public key cryptosystem // ANTS-III. 1998. P. 267—288.

И. В. Парфилов, П. А. Силин, Ю. Ю. Шумилов

ПОДХОД К РЕШЕНИЮ ПРОБЛЕМЫ ВЗАИМНЫХ БЛОКИРОВОК В МНОГОПОТОЧНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

При разработке сложных многопоточных программных средств важно учитывать ряд возможных проблем, которые в дальнейшем могут привести к их некорректной работе. Одним из признаков некорректной работы многопоточной программы является возникновение ситуации взаимной блокировки потоков в процессе ее выполнения, вызванное некорректным использованием средств синхронизации. Возникновение ситуаций взаимных блокировок может приводить к полной или частичной потере функциональности программных средств, в частности к появлению уязвимостей в системе безопасности. Проявление ситуаций взаимных блокировок сильно зависит от динамики выполнения программы в конкретной программной и аппаратной среде. Это свойство не позволяет создать сколько-нибудь эффективные эмпирические алгоритмы выявления взаимных блокировок на этапе тестирования программного обеспечения. В связи с этим на этапе проектирования многопоточной программы возникает проблема поиска потенциальных ситуаций взаимных блокировок.

Для выявления ситуаций взаимных блокировок на этапе детализированного проектирования используется метод проверки на модели. Такой подход имеет ряд преимуществ:

- он полностью автоматизирован, от пользователя лишь требуется провести моделирование проектируемой системы;
- если в проектируемой системе возможны ситуации взаимных блокировок, то в процессе проверки на модели всегда будет показана структура средств синхронизации, являющаяся результатом их некорректного использования и приводящая к возникновению ситуации взаимной блокировки.

В работе используется описательная модель многопоточного программного обеспечения. В качестве понятий, которыми оперирует модель, используется:

- объект — разделяемый ресурс (информационный или функциональный объект, к которому возможен доступ из разных потоков);
- субъект — поток, выполняющий доступ к разделяемому ресурсу;
- исключающий семафор — средство синхронизации, ограничивающее доступ к разделяемому ресурсу.

Основные преимущества описательной модели многопоточного программного обеспечения по сравнению с моделями, приведенными в [1, 2]:

- строгость, позволяющая в формальных математических терминах определять структуры, являющиеся результатом некорректного использования средств синхронизации;
- наглядность, позволяющая относительно легко исправлять некорректные структуры с целью избавления системы от ситуаций взаимной блокировки;
- полнота, позволяющая обрабатывать потоки произвольного вида (в терминах структурной теоремы Э. Дейкстры это означает, что субъекты могут содержать операторы ветвления и цикла [3]).