UDC 004.056.55

An overview of strategic workflow design to develop multimodal biometric recognition systems

Dhiman Karmakar (Department of Computer Science, Surendranath College)

It has been a proven fact that the inclusion of more than one modality makes biometric recognition systems more robust and elevates its recognition accuracy. This paper aims to develop a strategic platform for the upcoming researchers in order to develop multimodal biometric recognition systems. The researchers in this fraternity would be able to design their own procedural strategy using the generalized workflow template depicted in this review. The selection of proper visually interpreted biometric identifiers and modalities, various fusion strategies, choice of performance metrics, formation of training and test sets from databases and the possible challenges in developing the workflow are purposefully portrayed in this article.

Keywords: unimodal, multimodal, recognition, fusion, spoofing, metrics

1. Introduction

Nowadays, Biometrics, the physiological and behavioral traits used to recognize a human being, have become an integral part of human society with the increasing need for security at various levels in forensic, surveillance and commercial fields. A biometric security system should increase the security and secrecy of user data and be capable of authenticating a person’s identity based on his biometric traits like faces, fingerprints, iris etc.

Preference for face and face-like patterns for a child occurs hours after birth and perhaps starts with the face of the mother. According to the Intersensory Redundancy Hypothesis (IRH), during the early development of a child, perception of faces is enhanced in the unimodal visual (i.e., silent dynamic face) rather than bimodal audio-visual (i.e., dynamic face with synchronous speech in the form of lullaby) stimulation. In later days, this capability of a child is developed into the ability of face recognition of an adult by correlating many multimodal traits through expertise gained over the years. However, when the same task is undertaken by machine intelligence, the paradigm becomes altogether challenging and difficult. As human society is developing, increasing needs are felt for security systems that would identify a person from many of his traits and physical forms, such as identification cards, voice, gaits and gesture, and combination of many other parameters under varying conditions and situations.

The broad domain of Multi-Modal Biometrics (MMB) scenario works in either of the following modalities [1].

i. Multi-sensor: Different sensors (e.g. optical and/or electronic) used to capture the same biometric (e.g, fingerprints).

ii. Multi-biometrics: At least two different biometrics (say face and fingerprints) are fused and used for the purpose of recognition.

iii. Multi-unit: Multiple units captured from the same biometric (e.g. fingerprints collected from more than one finger).

iv. Multi-snap: Use of multiple instances of the same biometric (e.g. multiple impressions of the same finger).

v. Multi-matcher: Combining different approaches in feature selection and biometric matching algorithms [25].

A pictorial representation of the MMB scenario is shown in Fig.1. It is important to realize that a multi-biometric system is always multimodal, however the reverse is not true. It is worth noting that a single biometric may possess multiple modalities. For example, the geometric structure of a face may be blended with its behavioural nature to produce facial multimodality [27]. Similarly, finger veins and finger knuckles may be combined to produce multimodality in fingers [35].

Multimodal Biometric Systems (MMBS) have gained popularity among researchers as those provide more variability in information, for processing purposes of an individual rather than a unimodal biometric system [59, 46]. Those are useful to enhance the robustness in many security related areas including passport verification and authentication of persons.

2. Selection of authentication method in workflow

In this context, it is important to distinguish some frequently used terms in literature for the purpose of authentication, determining the identity of an individual in an automated manner. These terms are somewhat used inconsistently and interchangeably in different literature and may create confusion for the readers. Personnel can be authenticated using his (i) personal ID card (what he has), (ii) password (what he knows) and (iii) biometric traits, termed as recognition. Although a general approach to authenticate personnel is password matching or ID card verification, yet, such mechanisms of identity detection can easily be lost, hampered or stolen and thereby undermines the intended security. Incorporating physical and biological traits of human beings in the system has thus become a research issue in modern biometric security.

The process of recognition can further be divided into a couple of categories, namely, verification (“Is this the person who he claims to be?”) and identification (“Is this person in the database?”). It is quite obvious that verification is a 1:1 matching approach as the user’s data is verified only with the claimed person. Similarly, identification is a 1:N matching approach as the user’s identity is compared with all (N) the person’s data present in the database.

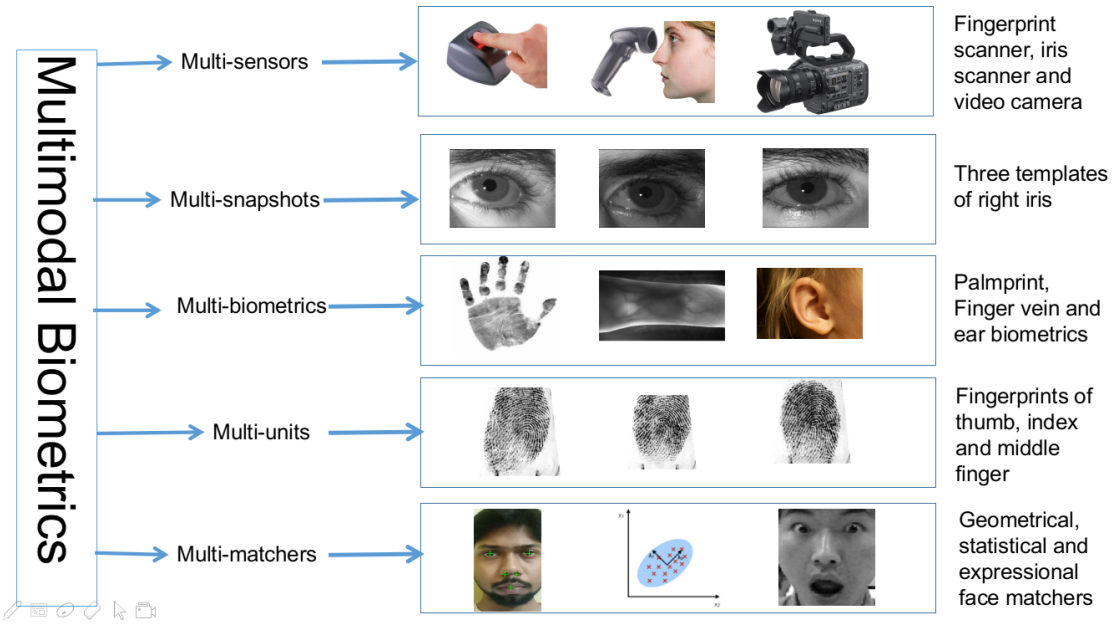


Figure 1: Various scenarios in a multimodal biometric system.

3. Selection of visually interpreted biometric identifiers

Biometric identifiers are categorized into physiological (say, fingerprint) or behavioral (say, gait) or a blend (say, voice) or some combinations. They can further be classified into three sections, namely hard, soft and hidden. The traditional feature-rich biometrics with self-sufficient ability to detect a human being are called hard biometrics. They are face, fingerprints, iris etc. Their level of accuracy is generally very high.

Soft biometrics, unlike the standalone nature of hard ones, are associated with classical biometrics to assist the overall recognition process. Skin color (related to face), height (related to full body) etc. are examples of soft biometrics. Though accuracy is relatively low, their nature of perceiving physiological traits (say, how tall or dark a person is) with ease, makes them popular. The hidden biometrics cannot be observed by naked eyes and generally they are stored in the form of medical data (for example, DNA report, blood group or X-ray image).

In this article, we confine the discussion to visually interpreted identifiers. They are the traits commonly represented in image or video forms and analyzed using computer vision techniques. For

this reason, DNA samples, odor etc. are not taken into consideration. Some of the traits, like speech data, though represented directly as signals, may be converted to image form for suitable analysis.

Face, fingerprint, palmprint, iris, ear and several others are the different identifiers used for the purpose of automatic human recognition. Each of them carries the pros and cons of their own. Seven characteristics [25], namely universality, distinctiveness, permanence, collectability, performance, acceptability and circumvention should be prioritized as per researcher's goal before the workflow design begins. For example, if permanence is assigned more priority than distinctiveness, then ear should be a preferred choice over face. Similarly, if performance is the key, iris can be an automatic alternative. However, choice of a biometric identifier largely depends upon the researcher's interest, research specific goal and availability of concerned databases.

4. Workflow pattern for a MMBs

MMBS are based on capturing human bodily features and using them for identification and authentication. MMBs function in an abstract manner. Generally, CBIR (Content Based Image Retrieval) system retrieves a query image and primitive features denoting image content, such as colour, textures, and shape, are computed for both stored and query images, and then used to identify stored images, most closely matching the query image. Semantic features such as the type of object present in the image, though difficult to extract, remains an active research area. Similarly, in a MMBs a query image, showing a biometric trait (say, face or fingerprint or iris etc.) is searched from a stored database of the same biometric. An overview of the generic MMBs is illustrated with the help of Fig.2. The entire procedure is divided into two sub-phases, namely training and testing.

4.1. Training phase in workflow

During the training phase, let us consider that the input database consists of multiple biometric modalities and is assumed to be structured and known. The term known indicates that the system and by structured we mean that every instance is properly placed within its class once enrolled. For example, every fingerprint of a person Xj should belong to class j, considering the presence of any person Xi in a database of n persons, where i = 1, 2, .., j, .., n.

The main algorithm of the system used to match a known and an unknown instance is called the matcher. The matcher is used to classify an unknown instance to its proper class. The efficacy of a matcher is adjudged by its score, crudely a number of correct classifications in comparison to wrong ones. The score can be viewed as the rate of recognition of the system. The different input

parameters of a matcher, for example, a global classification threshold value, can be adjusted at the end of each epoch of the algorithm in order to produce a better result. In other words, considering the input parameters and matcher as an encapsulated entity, the matcher is as a whole modified or trained to enhance the level of score.

The dataset used to train the classification algorithm is called the training dataset. Evidently, the term “trained dataset” would be wrong as the dataset is itself not trained and rather used to train the algorithm. Other than the training part the input database is divided into what is called a validation set, the dataset used to validate the matcher. The matcher takes an instance from the validation set and classifies it. Whether it is a mismatch or a correct classification can easily be determined, since the validation set is actually designed from the known database. Based on the result of classification, several input parameters are adjusted to enhance the system outcome. Here human interactions (not automated), for example, manual manipulation of classification threshold value may take place. The training phase concludes once the developer is satisfied with the system outcome. Therefore, the entire training phase helps to train the classification algorithm to a level of significant satisfaction.

4.2. Testing phase in workflow

In the testing phase, the unknown instances are fed to the modified matcher for classification. The database used to test the accuracy of the system is the test database. A robust MMBs should not only classify a test instance (closed set scenario) but also specify if the unknown instance does not fall into any of the existing known classes (open test scenario). In both the cases, the raw data captured is normalized into feature vector form and fusions are carried out at different levels. In the preprocessing phase, one may select the Region of Interest (RoI) of the image instance for further processing. Before feeding the data to the matcher, its dimensionality is reduced for faster processing and the required features (based on which the matcher is designed), generally in vector form, are extracted. Some of the feature extraction and dimensionality reduction schemes commonly used are PCA, KPCA, 2DPCA etc.

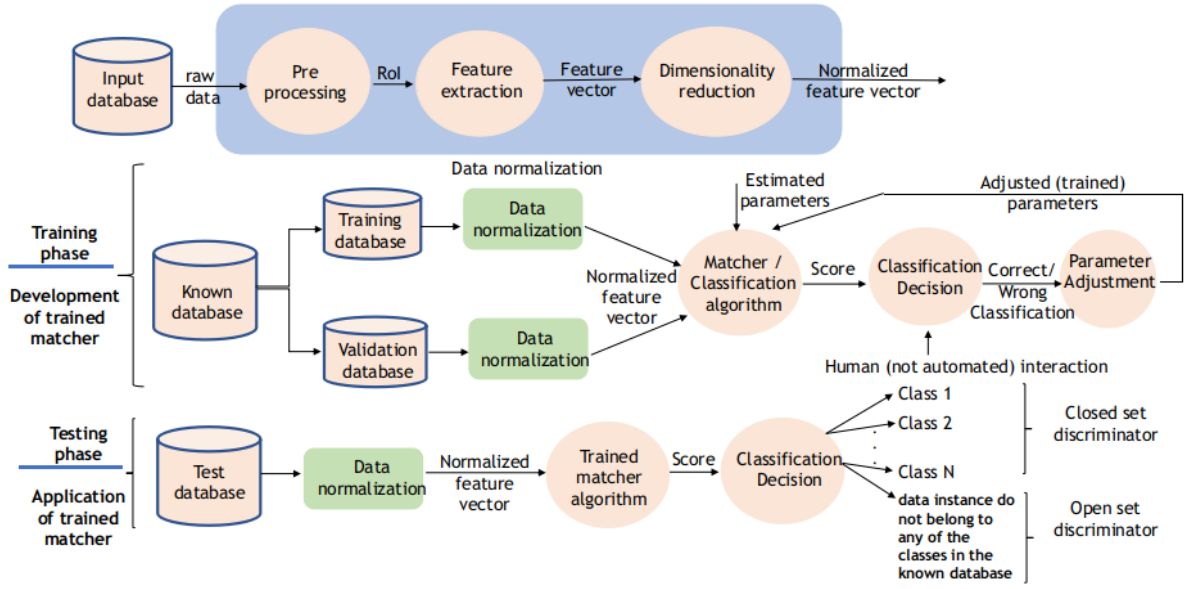


Figure 2: Abstract overview of a biometric recognition system

4.3. Workflow modes

The acquisition and processing in MMBs is carried out in three different modes.

i. Serial mode: Here, a biometric trait is processed before accepting the next one and the decisive classification outcome of the former is forwarded to the latter. It may reduce the overall recognition time as a decision can be drawn without accepting the next modality in sequence.

ii. Parallel mode: Here multiple modalities are processed simultaneously and the conclusion is drawn integrating the outcomes of all modalities.

iii. Hierarchical mode: Here individual matchers are combined in a treelike structure in order to manage several classification approaches.

5. Levels of information fusion in workflow

In MMB scenarios, how to fuse information of two different biometrics of the same person remains a matter of research. These fusion strategies can be performed in raw level, or in feature level, or in score level or even in decision level [24]. In the raw (or sensor) level of fusion, the same characteristics of raw MMB data captured using different sensors (for example, sensing the fingerprint of the same finger of a person in more than one fingerprint scanner) are combined. At times it is found to enhance the recognition accuracy but suffers from the drawback of incompatibility of data from different modalities.

Since the feature set of any biometric trait is supposed to carry the most significant and rich information about raw biometric data, the integration at feature level is expected to provide better recognition performance than the other levels of fusions. Concatenating the feature vectors extracted from the face and fingerprint data is an example of such fusion. Intuitively, the most important features of two biometric traits are expected to provide good performance. Although possessing these advantages, fusion in feature level is relatively understudied in comparison to other fusion levels [46]. The reasons are basically threefold. Firstly, extracted features from different biometric traits may become incompatible to each other; for example, extracted minutiae from a fingerprint and derived eigen coefficients from a face seem to be incompatible. Secondly, when different feature vectors of different biometrics are concatenated together, the resulting feature vector may suffer from the curse of dimensionality [32] and would become very difficult to handle. An effective dimensionality reduction scheme [53] for online authentication of face and signature, however, proves its significance. Thirdly, and most importantly, designing a matcher algorithm for a fused feature vector, containing features of different multimodal traits, is more difficult than generating separate matcher algorithms for different single biometrics. Fusion of information at this level also faces other challenges like large inter-user similarity, small intra-person variability, and unknown relationship between features [24].

A matcher algorithm, as discussed earlier, produces a similarity score based on the proximity of a query feature vector with the template feature vector in the known dataset. In score (or confidence or rank) level fusion, match-scores obtained from various matchers are combined for the final classification decision. The method of weighted average is often used to combine the scores. This level of fusion is attractive due to its simplicity and good performance [24].

Fusion at decision level is least powerful in comparison to other levels [46]. Here a separate decision is taken for each biometric at a very late stage and hence it prohibits enhancing the rate of recognition. A majority voter scheme is often applied to furnish the final decision on classification.

During the last decade several MMB verification and identification algorithms have been proposed including a wide variety of survey papers. Since the majority of the research work emphasizes modality in isolation, judging the efficacy of MMBs depends upon their (modalities) comparative analysis. This, however, is difficult to perform and may not provide a fruitful result.

Technologies are being developed for embedding multiple biometric information in the identity cards. For example, ICAO (International Civil Aviation Organization) encourages American people to use facial image, fingerprints and iris in their travel documents as a verification tool. Indian citizens are using Aadhaar as their identity card, which combines face, iris and fingerprint traits.

In MMB fusion strategy, at least two different biometrics are fused. Fusion at feature level, though difficult and understudied [46], possesses more importance than the other levels, because the extracted feature set from raw data holds most significant and rich information. Several evaluation protocols on closed test set identification have been designed for measuring the performance of different existing algorithms. In open test identification, the challenge is to reject the imposters. However, how to fuse information of two different biometrics of the same person remains a matter of research. The combined classifier approach has been adopted to get better results, especially at score level fusion.

In a brief review of MMB, Ross and Jain [46] have presented the idea of various levels of fusion, various possible scenarios, and different modes of operation, integration strategies and design issues. For homogeneous feature sets (e.g., multiple fingerprint impressions of a user finger), weighted average of the individual feature sets are often used to compute the resultant feature set (e.g., fusion using multiple hand features by Michael et al. [19]). On the other hand, for non-homogeneous feature sets (e.g., feature sets of different biometric modalities like face and hand geometry), we can concatenate them to form a single resultant feature set.

A combination of face and fingerprint authentication approaches using CNN (Convolutional Neural Network) for casting votes in elections through a web portal is explained by Saravanan et al. [3]. A MMB fusion scheme of face and fingerprints (specifically ridge and minutiae) [52] yields a recognition rate of 95%. Riseul et al. [47] designed a survey report towards continuous MMBs. The review has pointed out the deficiency of adequate number of comparisons between biometric types, fusion models and types of machine learning algorithm (supervised or semi supervised) in the published literature in this domain. On the contrary, a multimodal sparse technique of representing MMB data [55] by a scattered linear mixture of training records claims to yield better performance than traditional fusion schemes.

5.1. Feature level fusion

After formation of the MMB class of each person of the dataset, the feature level fusion is carried out on the biometrics. Ross and Jain [46], in a MMB review article elaborate different levels of fusion, integration strategies and design issues. Experimental results claim that MMB fusion improves both throughput and performance of the system. Rattani et al. [45] apply a feature level fusion of face and iris. Their algorithm computes the SIFT (Scale Invariant Feature Transform) features from the biometric sources. However, the method requires segmentation of the captured

images. Fortunately SIFT method can be experimentally combined with color image segmentation strategy for improved processing speed and better performance. Face and iris features may be extracted separately and fed into a wavelet probabilistic neural network classifier. The results are then calculated on the decision layer of the neural network. Nasrabadi et al. [56] present a deep learning approach to feature level fusion of face and iris. Their integration of deep hashing, a binarization technique, into the fusion architecture, generates robust MMBs. Utilization of modified gravitational search algorithm (GSA) in feature level fusion yields satisfactory results [23].

Finger based MMBs, being highly secured and stable, attract the attention of the research fraternity. Shuyi et al. [35] propose a MM fusion technique by combining finger veins with finger knuckle point patterns. This feature learning algorithm, maximizes the correlation between inter-modality samples. The problem has also been handled using CNN based approach by further inclusion of fingerprints with the aforementioned bimodal traits of fingers [36]. Sarangi et al. [50] develop a feature level fusion based MMB recognition of ear and profile face. The inclusion of face stabilizes the system in terms of its recognition rate, which may otherwise be hindered by the uncontrolled environment during ear enrollment. Construction of a unique template for each person, fusing his face, ear and palmprint data at feature level, and yet maintaining a relatively low-dimensional feature vector is discussed by Bokade et al. [5]. A multi-level fusion of unimodal methods with trimodal feature level fusion of face, iris and ear is proposed by Purohit et al. [43]. An accuracy of 95% in the trimodal case is found better than the highest achieved unimodal (iris) accuracy of 94%.

5.2. Fusion other than feature level

Some of the important work regarding multimodal dataset fusion strategies other than feature level are briefly summarized. Byeon et al. [6] proposed a deep learning model for multi modal biometric fusion, where they firstly used some fusion strategies at pixel level to optimize the process. Secondly, back propagation was used at feature level to establish relationships among the modalities and lastly, some intelligent fusion techniques were used at the score level. Claims are also made that MMBS security can be enhanced by subsequent use of early fusion, late fusion, and score-level fusion [2]. Performance of different fusion approaches, including image-level fusion, feature-level fusion, and two score-level fusion methods are also explored with the help of deep learning approach [22].

Choudhury et al. [12] apply a new framework of person authentication through adaptive rank level fusion. The proposed approach builds up a meta-heuristic design using ant colony optimization techniques on fingernail plates. A less explored combination of iris with palmprint

seems to be useful [57] in order to authenticate personnel by score level fusion. The article shows an interesting approach of bit transition code in Gabor filtered images. Inclusion of iris with face in uncontrolled condition may, however, worsen the system performance. A face-iris quality assessment network is proposed by Luo et al. [37] in order to decrease the effect of poor quality samples. Here the adaptive weights are also assigned to face and iris, based on their relative quality score and thereafter a score level fusion is applied to obtain satisfactory results. Being unobtrusive, soft biometrics, like gestures and postures are interesting choices in MMBs. Cherifi et al. [9] obtain an EER of 5.15% in score level fusion, where the user of a mobile phone is verified by his single action arm gesture while answering the phone. Ear shape structures are also extracted using the local phase quantization method. A normalized score level fusion strategy on iris and finger traits using hybrid genetic algorithm and Particle Swarm Optimization are applied by Sujatha et al. [54] to reduce FAR and FRR of the system. A score level fusion of fingerprints, finger-knuckle point and palmprint in a MMBs is illustrated by Kant et al. [28]. The suitability of cryptographic MMB authentication in medical application has been discussed by Mohsen et al. [14]. In this work, face and voice are used in both feature as well as score-level fusion.

5.3. Fusion of classifiers

The use of Artificial Neural Network, Artificial Intelligence, Genetic Algorithm etc. as tools in developing the classifiers or matchers has been attempted nowadays. Mikel et al. [34] clarifies the importance of MMB authentication for online student evaluation in COVID-19 pandemic scenario, by the development of an AI based procedure and thereafter testing it in a large scale system. ANN was used as a tool for MMB extensively. For example, Gokulkumari [21] traines the ANN using a modified dragonfly algorithm by selecting optimal weight in order to achieve classification accuracy. Rahman et al. [15] applied CNN successfully on ECG signal and fingerprint in both parallel and sequential models. CNN was also used to train the workflow model while integrating traits like Photoplethysmography and Electrocardiogram signals [13]. Rajasekar et al. [44] proposed a deep learning approach using CNN to integrate face, fingerprint and iris in MMBs.

A MMB recognition approach using the firefly algorithm of ONN (Optimal Neural Network) on fingerprint and ear is used by Chanukya et al. [8]. A median filter approach in preprocessing to identify the RoI of the given traits for further cropping is also applied. Alshardan et al. [2] embraced three renowned CNN architectures viz. ResNet, VGGNet, and DenseNet, to extract features from finger vein and fingerprint images.

In recent times, the combined classifier approach has been adopted to get better results, especially at score level fusion. That is, we may have different feature sets, different training sets, different classification methods or different training sessions, all resulting in a set of classifiers whose outputs may be combined, with the hope of improving the overall classification accuracy. A classifier combination is especially useful if the individual classifiers are largely independent. Various re-sampling techniques like bootstrapping may be used. Examples are stacking, bagging and boosting (or ARCing).

6. Performance metrics: The workflow verdict

The proper feature exaction and computationally efficient matching, clustering and classification algorithms make the MMB recognition systems reliable and robust. The efficiency of a MMBs is measured by its performance metrics which deal with the rates of success and failure (error) of the system. These metrics are expressed in terms of ratio or percentage or frequency. During the enrollment phase, the system may sometimes be unable to sense the biometric modalities (say, due to the lack of ridges in the fingerprint) of some users or fails to capture user's biometric data (say, due to the technological fault of the sensors). The former is known as *FTE* (Failure to Enroll) error while the latter *FTAR* (Failure to Acquire).

Biometric verification scenario may be viewed as a binary classification problem and can be analyzed with the help of a confusion matrix [33]. Two types of system errors are encountered in the verification process.

i. Type-I error or *FRR* (False Rejection Rate), which indicates the proportion of genuine users falsely predicted as impostors and hence rejected to grant the system access. It is expressed as below.

$$FRR = \frac{Total\ false\ rejection}{Total\ true\ attempts}$$

This error is also commonly termed as False Non-Match Rate or False Negative Rate.

ii. Type-II error or *FAR* (False Acceptance Rate), which signifies the proportion of impostors falsely predicted as genuine users and being accepted to grant the system access. Its expression is shown below.

$$FAR = \frac{Total\ false\ acceptance}{Total\ false\ attempts}$$

This error is also commonly termed as False Match Rate or False Positive Rate.

iii. Analogously, True Rejection Rate (*TRR*) and True Acceptance Rate (*TAR*) are used as two success rates of the system. In an ideal MMB verification scenario, FAR = 0, FRR = 0, TAR = 1 and TRR = 1.

$$TRR = \frac{Total\ true\ rejection}{Total\ false\ attempts}$$

$$TAR = \frac{Total\ true\ acceptance}{Total\ true\ attempts}$$

iv. The thresholds set in a MMBs trivially estimating the training set result and test set result are termed as a *priori* and a *posteriori* respectively. A posteriori threshold can be considered as a finally adjusted parameter in the training phase of Fig.2. Other than soft and hard, adaptive threshold [26] has its own significance in biometric recognition. Ideally, if a MMBs matcher maintains a very low threshold, there will be maximum acceptance (or minimum rejection) irrespective of true or false attempts. Similarly, a maximum rejection (or a minimum acceptance) will occur for a very high threshold value. Since FAR and FRR are inversely related, no adjustment of threshold can decrease them simultaneously. The error percentage (in terms of FAR and FRR) against threshold is plotted. To maintain a trade-off between such a high and low threshold value, the intersecting point of FAR and FRR, where both the error rates are equal, is estimated as the threshold value. This point is termed as Equal Error Rate (*EER*) or Crossover Error Rate (*CER*).

v. The Receiver Operating Characteristic (*ROC*) curve (uses the normal deviate scale) or Detection Error Trade-off *(DET*) curve (uses linear, semi-logarithmic or logarithmic scale) and designed to measure the performance of a classification model. ROC serves as another ploy to detect the EER. Here, FAR vs FRR is plotted and the point on the curve, where FAR equals FRR signifies EER. Obviously, lower the EER, better is the system.

vi. Some of the MMBs prefer to use Half Total Error Rate (*HTER*), the averaged value of FAR and FRR as their performance measure, as indicated below.

$$\text{HTER} = \frac{FRR + FAR}{2}$$

However, *HTER* can only impose a gross effect on system efficacy and is not an issue for a MMBS of specific purpose. For example, a MMB whose security is the main concern, given two systems of equal *HTER* should opt for the one with lesser FAR.

vii. The system performance in the identification scenario is measured in terms of its accuracy or Recognition Rate (RR), as stated below.

$$RR = \frac{total\ correctly\ identified\ sample}{total\ test\ sample}$$

The correct or wrong identification of a test sample is decided as follows. A test sample x_i out of total n test samples x_1 , x_2 , ..., x_n is fed to a matcher $m(x_i , r)$ of rank r. The matcher returns r most closely matched samples of xi from the training dataset, out of which k are truly and $(r - k)$ are falsely matched. The rank value, $rank_r = \frac{100k}{r}\%$, more than a threshold percentage $\theta\%$, portrays true identification. For example, given $r = 10$ and $\theta = 80$, say the matcher returns $k = 9$ true matches resulting $rank = 90\%$ and since $90 > \theta$, the match (identification) is concluded to be correct. A plot of $rank_r$ vs $r = 1, 2, ...n$, known as *CMC* (Cumulative Match Characteristic) curve yields a summarization of the identification effectiveness. The proportion of test samples misclassified to a wrong bin (any class other than where it originally belongs to) termed as *bin error rate* or misclassification rate signifies the failure rate in identification.

viii. The duration of the matching process from the end of the enrollment phase until the classification decision, is called Time to Match (*TTM*) rate and is often used as a final conclusive way of estimating the performance of a MMBs.

Spoofing attacks are often responsible for downgrading or even compromising MMBs performance, which emerges from the necessity of anti-spoofing systems. Safavipour et al. [48] claimed that multimodal templates obtained from their deep learning strategy of feature spaces are extremely secure against spoof attacks. A cancellable MMBs capable of protecting the actual biometric features from the intruders was developed by Umer et al.[51]. The security of online and IoT-enabled authentication was also maintained through a method of encryption-decryption. Liveness detection is a modern anti spoofing technique, where the system diagnoses whether the impersonation is caused by representing a fake biometric sample instead of the actual live human being. Dhiman et al. developed two unique liveness detection approaches, namely, multivariate gradient descriptor and multi dimensional Fourier transform applied on facial micro-expression regions [29, 31].

A self-explanatory tabular form summarizes some state-of-the-art MMBs methods for researchers' benefit (see Table 1). The identifiers fused, their levels of fusion and the keynotes in the proposed workflow model are depicted in separate columns.

Table 1: State-of-the-art MMBs methods in a nutshell

Ref.	**Biometrics**	**Fusion level**	**Key points in workflow design**

[2]	Fingerprints finger and veins	Early, late and score	Claims were made that subsequent levels of fusion could make a MMB more robust.
[3]	Face and fingerprints	Decision	CNN based voting system in web portal.
[5]	Face, ear and palmprint	Feature	Feature vector was restructured even after considering three biometric traits, to lesser dimension.
[6]	Face, iris and fingerprints	Pixel, feature and score	Used intelligent fusion techniques through deep learning.
[8]	Fingerprint and ear	Feature	Firstly a median filter was used to extract the RoI and thereafter firefly algorithm of ONN was applied for recognition.
[9]	Ear and arm	Score	User of a mobile phone was verified by his arm gesture and ear portion.
[12]	Index, middle and ring fingernails plates	Rank	The work contributed towards optimal performance accuracy using ant colony optimization and deep learning.
[13]	PPG and ECG signals	Feature and score	Federated learning, an efficient machine learning approach, was used to collaborate decentralized modules without effective data exchange, for security management.
[14]	Face and voice	Feature and score	Established the importance of cryptographic MMB authentication in medical imagery.
[15]	ECG and fingerprints	Decision and score	Deep learning and traditional classification module were used to evaluate the proposed system.
[22]	Face and iris	Pixel, feature and score	An effective deep learning approach was used in the extracted region of interest of the images.
[23]	Face, iris and fingerprints	Feature	Used a threshold specific optimization technique in gravitational search algorithm to outperform the traditional methods.
[28]	Fingerprints, finger knuckle and palmprint	Score	The resulting match score was used to detect authenticity by comparing different performance metrics.
[34]	Face, audio and	Decision	An AI driven monitoring system for online

	keystroke dynamics		student evaluation.
[35]	Finger veins and finger knuckle	Feature	Portrayed multimodality in unibiometrics (finger) and maximized the correlation between inter-modality samples.
[36]	Fingerprints, finger veins and finger knuckle	Feature	Trimodal representation of uni biometric (finger).
[37]	Face and iris	Feature	Used generalized divisive normalization and assigned adaptive weights in image samples to reduce poor image acquisition quality.
[43]	Face, iris and ear	Feature	Recognition rate for unimodal and multimodal data were compared and contrasted in detail.
[44]	Face, fingerprints and iris	Feature and score	CNN was used in conjunction with softmax classifier for identification purposes.
[48]	Face, iris and fingerprints	Feature	Feature vectors were mapped from the feature space into the reproducing kernel and deep learning combined them in fully connected in-depth layers.
[50]	Ear and face	Feature	Uncontrolled enrollment of ear was stabilized by the inclusion of face traits.
[51]	Face, iris and palm prints	Score	A cancelable biometric system (CBS) was introduced to preserve the original traits from possible external misuse.
[52]	Face and fingerprints	Feature	Used variable crossing number(CN) which determines minutiae and ridge ending or bifurcation.
[53]	Face and signature	Feature	An online authentication system with novel dimension reduction approach.
[54]	Iris and finger	Score	Hybrid genetic algorithm and Swarm optimization were used to enhance the recognition rate.
[56]	Face and iris	Feature	Integrated the learning approach of deep hashing in fusion strategy.
[57]	Iris and palmprints	Score	Used bit transition codes in Gabor filter images.

7. Database selection in workflow

It has been observed in the recent past that inclusion of more than one modality in the image dataset, though increases the data handling and manipulation effort, drastically improves the efficiency of the system in terms of its rate of recognition. In this context, the terms database and dataset are interchangeably used. However, specifically, a database is an organized collection of data stored as multiple datasets which are in turn storage of structured data for a specific purpose.

For a given query, which can be an image in case of iris, face, fingerprint and palmprint recognition problems or can also be a signal in case of speech data, the system should be able to conclude whether it matches with any in its database. The system should have the option of deciding that the query does not belong to the given data set, and hence classify it as an imposter.

Some popular databases for faces are FIA [20], YALE [18], ORL [49], FERET [41] etc. and for iris are MMU [39], CASIA [10], DOBES [38] etc. FIA and YALE datasets consist of frontal faces with varying facial expressions while FERET contains faces with different angular directions. A collection of A-Z face database repositories is collectively uploaded in Princeton University's webpage [42]. NIST [16] and CASIA [10] maintain databases containing various forms of fingerprint data for research purposes. SVC 2004 [58] is one among the very few online signature databases publicly available to the research community. Including the well-known corpora [7] many other databases are freely available in case of speaker recognition.

Multimodal databases are not as widely obtainable as unimodals. To handle such scarcity, researchers either create their own database, which is typically effort-heavy, or may opt for the following interesting approach. If two different traits are uncorrelated (features of one trait does not depend on another, for the same person), like face and iris, identifiers obtained from different sources may be assumed as the same person's data [30]. However, for correlated traits, like ear and skin-color, such assumption is inane. In multimodal scenarios, the databases available are BIOMET [17] (face, hand, fingerprint, voice and signature), BANCA [4] (face and voice), MYCT [40] (fingerprint and signature) etc.

8. Conclusion in workflow design: challenges and benefits

Some of the major problems commonly come across in MMB research scenarios are enlisted below.

i. Class-variability: In vision-based MMB recognition, the presence of a high degree of variability in human biometric information has been a major concern. In other words, there exists an

infinite number of possible classes in a biometric database. There can be potentially very large intra-class variations and also rather small inter-class variations (due to the similarity of individual appearances).

ii. Noisy data: The presence of noisy and distorted data in the captured sample and thereby production of degraded quality samples is causing database enrollment error. This may further lead to a failure in identification algorithm and thereby arrival at a faulty decision.

iii. Spoofing attack: In order to get unauthorized access in MMBs, biometric spoofing has often been applied to compromise the system. To combat such attacks, researchers are developing various anti-spoofing technologies.

iv. Population coverage: The lack of population coverage is another challenge faced by the system designers to construct proper databases. However, construction of large sample sizes manifests big data problems. Naturally, the sensed data is stored in compressed encrypted form for effective space utilization and speedy probing. However, such a storing scheme makes it difficult to retrieve the data in originally captured lossless form.

v. Social acceptance: Social issues pose barriers in the acquisition level of a MMBS. For example, face may be considered more user friendly than iris as the former requires less human interaction. On the contrary, due to the same reason the former can be captured without human intervention and causes privacy threats. A common threat in social media, function creep, where the captured traits are not utilized for the legitimate purpose, is difficult to restrain.

This review article offers several benefits to the emergent researchers in the field of MMBS, enlisted below.

i. Substantive amount of referred research articles to foster the circumstantial workflow designing.

ii. Choice of biometric identifiers as per researcher’s need.

iii. Selection of multimodal databases or formation of multimodal databases using different unimodal sources.

iv. Detailed comparison of fusion strategies to design the workflow.

v. Selection of a particular classification algorithm or developing the classifier fusion strategy in workflow.

vi. The explicit choice of authentication method and subsequently arriving at final verdict using proper performance metrics.

References

1. Afaneh A., Alqaralleh E., Toygar O. Person identification using multimodal biometrics under different challenges // In: Person Identification Using Multimodal Biometrics Under Different Challenges. 2018. Chapter 5. P. 81–100.
2. Alshardan A., Kumar A., Alghamdi M., Maashi M., Alahmari S., Alharbi A. A. K., Almukadi W., Alzahrani Y. Multimodal biometric identification: leveraging convolutional neural network (CNN) architectures and fusion techniques with fingerprint and finger vein data // PeerJ Computer Science. 2024. Vol. 10. Article e2440.
3. Arputhamoni S., Jireh J., Sarvanan A. G. Online smart voting system using biometrics based facial and fingerprint detection on image processing and CNN // Proc. of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). IEEE, 2021.
4. Bailly-Bailliere E., Bengio S., Bimbot F., Hamouz M., Kittler J., Mariethoz J., Matas J., Messer K., Popovici V., Poree F., Ruiz B., Thiran J. P. The BANCA database and evaluation protocol // Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA). LNCS 2688. 2003. P. 625–638.
5. Bokade G. U., Kanphade R. D. A novel approach for secured multimodal biometric authentication based on data fusion technique // International Journal of Computational Vision and Robotics. 2021. Vol. 11, no. 2. P. 214–243.
6. Byeon H., Raina V., Sandhu M., Shabaz M., Keshta I., Soni M., Matrouk K., Singh P. P., Lakshmi T. R. V. Artificial intelligence-enabled deep learning model for multimodal biometric fusion // Multimedia Tools and Applications. 2024. P. 1–24.
7. Campbell J., Reynolds D. Corpora for the evaluation of speaker recognition systems // Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’99). 1999. Vol. 2. P. 829–832.
8. Chanukya P. S. V. V. N., Thivakaran T. K. Multimodal biometric cryptosystem for human authentication using fingerprint and ear // Multimedia Tools and Applications. 2020. Vol. 79, no. 1. P. 659–673.
9. Cherifi F., Amroun K., Omar M. Robust multimodal biometric authentication on IoT device through ear shape and arm gesture // Multimedia Tools and Applications. 2021. Vol. 80, no. 10. P. 14807–14827.

10. Chinese Academy of Sciences' Institute of Automation (CASIA) fingerprint dataset. Available at: http://biometrics.idealtest.org/findTotalDbByMode.do?mode=Fingerprint#/ (accessed June 05, 2025).
11. Chinese Academy of Sciences' Institute of Automation (CASIA) iris dataset. Available at: http://biometrics.idealtest.org/ (accessed June 05, 2025).
12. Choudhury S. H., Kumar A., Laskar S. H. Adaptive management of multimodal biometrics—a deep learning and metaheuristic approach // Applied Soft Computing. 2021. Vol. 106. Article 107344.
13. Coelho K. K., Tristão E. T., Nogueira M., Vieira A. B., Nacif J. A. M. Multimodal biometric authentication method by federated learning // Biomedical Signal Processing and Control. 2023. Vol. 85. Article 105022.
14. El-Bendary A. M., Mohsen H., Kasban H., Haggag A. Investigating of nodes and personal authentications utilizing multimodal biometrics for medical application of WBANs security // Multimedia Tools and Applications. 2020. Vol. 79, no. 33. P. 24507–24535.
15. El Rahman S. A., Alluhaidan A. S. Enhanced multimodal biometric recognition systems based abstract: on deep learning and traditional methods in smart environments // PLOS ONE. 2024. Vol. 19, no. 2. Article e0291084.
16. Fiumara G., Flanagan P., Grantham J., Bandini B., Ko K., Libert J. NIST Special Database 300 uncompressed plain and rolled images from fingerprint cards. National Institute of Standards and Technology. DOI: 10.18434/T4/1502472. Available at: https://doi.org/10.18434/T4/1502472 (accessed December 17, 2019).
17. Garcia-Salicetti S., Beumier C., Chollet G., Dorizzi B., Leroux les Jardins J., Lunter J., Ni Y., Petrovska-Delacretaz D. BIOMET: A multimodal person authentication database including face, voice, fingerprint, hand and signature modalities // Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA). LNCS 2688. 2003. P. 845–853.
18. Georghiades A. S., Belhumeur P. N., Kriegman D. J. From few to many: Illumination cone models for face recognition under variable lighting and pose // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. Vol. 23, no. 6. P. 643–660.
19. Goh K. O. M., Tee C., Teoh A. B. J. A contactless biometric system using multiple hand features // Journal of Visual Communication and Image Representation. 2012. Vol. 23, no. 7. P. 1068–1084.

20. Goh R., Liu L., Liu X., Chen T. The CMU face in action (FIA) database // In: Zhao W., Gong S., Tang X., editors. Analysis and Modelling of Faces and Gestures. Springer Berlin Heidelberg, 2005.
21. Gokul Kumari G. Metaheuristic-enabled artificial neural network framework for multimodal biometric recognition with local fusion visual features // The Computer Journal. 2021.
22. Hattab A., Behloul A. Face-iris multimodal biometric recognition system based on deep learning // Multimedia Tools and Applications. 2024. Vol. 83, no. 14. P. 43349–43376.
23. Ipeayeda F. W., Oyediran M. O., Ajagbe S. A., Jooda J. O., Adigun M. O. Optimized gravitational search algorithm for feature fusion in a multimodal biometric system // Results in Engineering. 2023. Vol. 20. Article ID 101572.
24. Jain A. K., Nandakumar K., Ross A. Score normalization in multimodal biometric systems // Pattern Recognition. 2005. Vol. 38. P. 2270–2285.
25. Jain A. K., Ross A., Prabhakar S. An introduction to biometric recognition // IEEE Transactions on Circuits and Systems for Video Technology. 2004. Vol. 14, no. 1. P. 4–20.
26. Joshi V. B., Raval M. S. Adaptive threshold for fingerprint recognition system based on threat level and system load // Procedia Computer Science. 2020. Vol. 171. P. 498–507.
27. Kakadiaris I. A., Passalis G., Theoharis T., Toderici G., Konstantinidis I., Murtuza N. Multimodal face recognition: combination of geometry with physiological information // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. Vol. 2. P. 1022–1029.
28. Kant C., Chaudhary S. A multimodal biometric system based on finger knuckle print, fingerprint, and palmprint traits // In: Innovations in Computational Intelligence and Computer Vision. Springer, Singapore, 2021. P. 182–192.
29. Karmakar D., Mukherjee P., Datta M. Spoofed facial presentation attack detection by multivariate gradient descriptor in micro-expression region // Pattern Recognition and Image Analysis. 2020. Vol. 31, no. 2. P. 285–294.
30. Karmakar D., Murthy C. A. Generation of new points for training set and feature-level fusion in multimodal biometric identification // Machine Vision and Applications. Springer. 2013. Vol. 24, no. 6. July.
31. Karmakar D., Sarkar R., Datta M. Spoofed replay attack detection by multidimensional Fourier transform on facial micro-expression regions // Signal Processing: Image Communication. Elsevier. 2021. Vol. 93.

32. Keogh E., Mueen A. Curse of dimensionality // Encyclopedia of Machine Learning and Data Mining. 2017. P. 314–315.
33. Kulkarni A., Chong D., Batarseh F. A. Foundations of data imbalance and solutions for a data democracy // Data Democracy. Academic Press, 2020. P. 83–106.
34. Labayen M., Vea R., Flórez J., Aginako N., Sierra B. Online student authentication and proctoring system based on multimodal biometrics technology // IEEE Access. 2021. Vol. 9. P. 72398–72411.
35. Li S., Zhang B., Fei L., Zhao S. Joint discriminative feature learning for multimodal finger recognition // Pattern Recognition. 2021. Vol. 111. Article ID 107704.
36. Li S., Zhang B., Zhao S., Yang J. Local discriminant coding based convolutional feature representation for multimodal finger recognition // Information Sciences. 2021. Vol. 547. P. 1170–1181.
37. Luo Z., Gu Q., Su G., Zhu Y., Bai Z. An adaptive face-iris multimodal identification system based on quality assessment network // In: International Conference on Multimedia Modeling. Springer, Cham, 2021. P. 87–98.
38. Machala L., Dobes M. Upol iris image database, 2008. Available at: http://phoenix.inf.upol.cz/iris/ (accessed June 05, 2025).
39. Multimedia University. MMU1 and MMU2 iris image databases, 2008. Available at: https://www.kaggle.com/datasets/naureenmohammad/mmu-iris-dataset (accessed June 05, 2025).
40. Ortega-Garcia J., Fierrez-Aguilar J., Simon D., Gonzalez J., Faundez Zanuy M., Espinosa V., Satue A., Hernaez I., Igarza J. J., Vivaracho C., Escudero D., Moro Q. I. MCYT baseline corpus: A bimodal biometric database // IEE Proceedings - Vision, Image and Signal Processing. 2003. Vol. 150, no. 6. P. 395–401.
41. Phillips P. J., Wechsler H., Huang J., Rauss P. J. The FERET database and evaluation procedure for face-recognition algorithms // Image and Vision Computing. 1998. Vol. 16, no. 5. P. 295–306.
42. Princeton University Library. Face image databases. Available at: https://libguides.princeton.edu/facedatabases#s-lg-box-27701497 (accessed June 05, 2025).
43. Purohit H., Ajmera P. K. Multimodal multilevel fusion of face ear iris with multiple classifications // In: International Conference on Modelling, Simulation and Intelligent Computing. Springer, Singapore, 2020.
44. Rajasekar V., Saracevic M., Hassaballah M., Karabasevic D., Stanujkic D., Zajmovic M., Tariq U., Jayapaul P. Efficient multimodal biometric recognition for secure authentication

based on deep learning approach // International Journal on Artificial Intelligence Tools. 2023. Vol. 32, no. 03. Article 2340017.

45. Rattani A., Tistarelli M. Robust multi-modal and multi-unit feature level fusion of face and iris biometrics // University of Sassari - Computer Vision Laboratory. (CVL 2008/003). Technical Report, 16 Dec. 2008. P. 15.
46. Ross A., Jain A. K. Multimodal biometrics: An overview // Proc. of 12th European Signal Processing Conference (EUSIPCO). Vienna, Austria, Sept. 2004. P. 1221–1224.
47. Ryu R., Yeom S., Kim S. H., Herbert D. Continuous multimodal biometric authentication schemes: A systematic review // IEEE Access. 2021. Vol. 9. P. 34541–34557.
48. Safavipour M. H., Doostari M. A., Sadjedi H. Deep hybrid multimodal biometric recognition system based on features-level deep fusion of five biometric traits // Computational Intelligence and Neuroscience. 2023. Vol. 2023, no. 1. Article 6443786.
49. Samaria F. S. ORL database of faces. Face recognition using hidden Markov models, Doctoral dissertation. University of Cambridge, 1994.
50. Sarangi P. P., Nayak D. R., Panda M., Majhi B. A feature-level fusion based improved multimodal biometric recognition system using ear and profile face // Journal of Ambient Intelligence and Humanized Computing. 2021. P. 1–32.
51. Saiyed U., Sardar A., Rout R. K., Tanveer M., Razzak I. IoT-enabled multimodal biometric recognition system in secure environment // IEEE Internet of Things Journal. 2023.
52. Singh L. K., Khanna M., Garg H. Multimodal biometric based on fusion of ridge features with minutiae features and face features // International Journal of Information System Modeling and Design (IJISMD). 2020. Vol. 11, no. 1. P. 37–57.
53. Singhal M., Shinghal K. Secure deep multimodal biometric authentication using online signature and face features fusion // Multimedia Tools and Applications. 2024. Vol. 83, no. 10. P. 30981–31000.
54. Sujatha E., Sundar J. S. J., Deivendran P., Indumathi G. Multimodal biometric algorithm using iris, finger vein, fingerprint with hybrid GA, PSO for authentication // In: Data Analytics and Management. Springer, Singapore, 2021. P. 267–283.
55. Suntharam V. S., Chandu R., Rajan D. P. Robust multimodal biometric recognition based on joint sparse representation // ICCCE. Springer, Singapore, 2021. P. 265–274.
56. Talreja V., Valenti M. C., Nasrabadi N. M. Deep hashing for secure multimodal biometrics // IEEE Transactions on Information Forensics and Security. 2020. Vol. 16. P. 1306–1321.

57. Vyas R., Kanumuri T., Sheoran G., Dubey P. Accurate feature extraction for multimodal biometrics combining iris and palmprint // Journal of Ambient Intelligence and Humanized Computing. 2021. P. 1–9.
58. Yeung D. Y., Chang H., Xiong Y., George S., Kashi R., Matsumoto T., Rigoll G. SVC2004: First international signature verification competition // Proceedings of the 2004 Biometric Authentication: First International Conference (ICBA 2004), Hong Kong, China. 2004. P. 16–22.
59. Waleed D., Fadewar H. S. Multimodal biometric system: A review // International Journal of Research in Advanced Engineering and Technology. 2018. Vol. 4. P. 25–31.

УДК 004.052.42

Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра»*

Кондратьев Д.А. (Институт систем информатики им. А.П. Ершова СО РАН)

Бояндин Л.К. (Новосибирский государственный университет)

Гончар Г.Е. (Новосибирский государственный университет)

Марченко В.В. (Московский государственный университет им. М.В. Ломоносова, Институт системного программирования им. В.П. Иванникова РАН)

Обухова А.А.

Разбитнова Ю.Ю. (Новосибирский государственный университет)

Хованская А.С.

Янбулатов Д.Р. (Новосибирский государственный университет)

*при поддержке Математического центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2025-349

Обеспечение доверия к реализациям криптостойких алгоритмов является актуальной задачей современного программирования. К правильности работы таких программных систем предъявляются повышенные требования, поэтому для доказательства корректности таких программ относительно спецификаций применяют дедуктивную верификацию. В данной статье описана работа в прогрессе по доказательству корректности реализации хэш-функции «Стрибог» из ядра Linux относительно ГОСТ Р 34.11-2012. Данная работа стартовала на проекте «Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра»» на Большой Математической Мастерской 2025 года. В качестве результатов работы мы презентуем формализацию ГОСТ Р 34.11-2012 в системе интерактивного доказательства Rocq. Данная формализация является функциональной спецификацией любой реализации хэш-функции «Стрибог». Также мы презентуем задание спецификаций для базовых функций реализации хэш-функции «Стрибог» из ядра Linux и методы упрощения доказательства таких функций с помощью задания набора лемм о связи структур данных из функциональной спецификации и из данной реализации.

Ключевые слова: дедуктивная верификация, логика Хоара, функциональная спецификация, хэш-функция «Стрибог», ГОСТ Р 34.11-2012, Rocq, Verified Software Toolchain

1. Введение

В настоящее время хэш-функции активно применяются в программных системах, где ошибки недопустимы, например, системах управления правами доступа,
блокчейн-системах, системах электронной цифровой подписи и т.д. Поэтому, актуальной задачей является обеспечение доверия к реализациям хэш-функций. ГОСТ Р 34.11-2012 [4] установил текущий стандарт хэширования в Российской Федерации, основанный на хэш-функции «Стрибог», где для вычисления хэша каждой 512-битовой части входных данных применяется функций сжатия на базе преобразования XLPS, в котором используются xor (X), линейное преобразование (L), подстановка (P) и нелинейное преобразование (S) [3, 24, 37, 54]. Особенно важно проверить корректность реализации хэш-функции «Стрибог» из ядра Linux [27], активному применению которой в российском программном обеспечении способствуют наличие отечественного стандарта ГОСТ Р 34.11-2012 на данную функцию и широкое внедрение в России операционных систем на базе ядра Linux, например, операционной системы Astra Linux [14], разработанной Группой Астра. Так как к корректности реализаций подобных функций предъявляются повышенные требования, то в таких случаях тестирования не достаточно. Для доказательства корректности программ применяется формальная верификация [1, 2, 8, 12, 25]. На первоначальном этапе верифи-

кации важно проверить корректность программы относительно функциональных свойств, описывающих взаимосвязь между входными и выходными данными. Для доказательства корректности программ относительно функциональных свойств применяется такой вид формальной верификации, как дедуктивная верификация программ [11, 13, 17, 30, 33].

Дедуктивная верификация основана на логике Хоара [22, 23, 31, 35], позволяющей свести задачу проверки корректности программ относительно спецификаций к задаче доказательства формул, называемых условиями корректности программы. В качестве спецификаций используются предусловия, являющиеся ограничениями на входные данные программы, и постусловия, представляющие собой ограничения на выходные данные программы и на связь входных и выходных данных программы. Упомянутое сведение осуществляется на основе правил из логики Хоара для всех конструкций языка программирования. Такой набор правил называется аксиоматической семантикой языка программирования. На базе аксиоматических семантик реализуют системы дедуктивной верификации программ, в которых для доказательства условий корректности применяют системы интерактивного доказательства теорем [34].

Важным видом аксиоматической семантики является прямое прослеживание [28], которое основано на применяемых к самой первой инструкции в программе правилах, что позволяет при выводе условий корректности продвигаться по инструкциям программы в порядке их исполнения. В случае, когда инструкция программы содержит работу с памятью через указатели, применяется такой специальный вид логики Хоара, как сепарационная логика [36, 49, 50, 52, 53]. Такая логика позволяет упростить работу с памятью с помощью разделения памяти на отдельные непересекающиеся участки и независимую работу с такими участками.

В данной статье описана работа в прогрессе по проекту дедуктивной верификации реализации хэш-функции «Стрибог» из ядра Linux. В качестве верифицируемой функциональной спецификации используется ГОСТ Р 34.11-2012. Отметим, что на текущем этапе проекта мы не верифицируем криптографические свойства [15, 18, 32, 42, 57] из-за приоритетности верификации функциональных свойств. Доказательство корректности выполняется с помощью системы дедуктивной верификации Verified Software Toolchain (VST) [20, 21, 26], основанной на погружении программы и ее спецификаций в среду интерактивной системы доказательства теорем Rocq (ранее называлась Coq) [51]. Для генерации условий корректности в системе VST используется прямое прослеживание и

сепарационная логика.

Данная работа выполняется на проекте «Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра»» на Большой Математической Мастерской 2025 года. Заказчиком данного проекта является Тимофей Юрьевич Черганов из Группы Астра. Куратором проекта является Дмитрий Александрович Кондратьев, к.ф.-м.н., научный сотрудник ИСИ СО РАН и старший преподаватель НГУ. В команду проекта входят авторы статьи, а именно:

- Бояндин Лев Константинович
- Гончар Глеб Евгеньевич
- Марченко Вадим Витальевич
- Обухова Алиса Андреевна
- Разбитнова Юлия Юрьевна
- Хованская Анна Станиславовна
- Янбулатов Денис Ринатович

Таким образом, вместе с куратором, проект выполняют восемь исполнителей. Коллектив проекта представлен на рисунке 1. Дополнительной задачей проекта является создание коллектива, способного выполнять задачи по дедуктивной верификации С-программ.

Основная задача проекта состоит в формальной верификации реализации хэш-функции «Стрибог» из ядра Linux и разработка подхода к формальной верификации реализаций криптографических функций. В данной статье мы презентуем такие результаты нашей работы, как формализацию ГОСТ Р 34.11-2012 в системе Rocq и задание лемм для теории предметной области, что может быть применимо в других подобных исследованиях. Результаты выполнения работы общедоступны в репозитории проекта [55].

Обзор родственных работ В качестве основной родственной работы отметим статью про дедуктивную верификацию хэш-функции SHA-256 в системе VST [19]. В данной работе, также как и в нашем исследовании, верифицировались не криптографические, а функциональные свойства, заданные стандартом на хэш-функцию. Однако из-за разницы стандартов в нашем случае в системе Rocq потребовалось задавать новые типы данных для хранения битовых векторов. Отметим, что мы использовали более гибкое представление для битовых векторов, параметризованное по их длине.

Также в качестве родственного исследования рассмотрим работы по формальной ве-

Рис. 1. Коллектив проекта «Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра»»

рификации такой части операционной системы Astra Linux, как система управления правами доступа [5–7]. Данные работы основаны на таком методе формальной верификации, как проверка моделей (model checking), который основан на проверке выполнение свойств на модели, описывающей состояния программной системы и переходы между ними. С одной стороны, такой подход, в отличие от дедуктивной верификации, позволяет автоматически проводить формальную верификацию. С другой стороны, в отличие от дедуктивной верификации, используемая при таком подходе абстракция системы не позволяет верифицировать низкоуровневые детали в виде реализации лежащих в основе хэш-функций. Для дедуктивной верификации частей операционной системы Astra Linux применяют систему дедуктивной верификации AstraVer [9, 29, 43, 45, 56]. Но используемые в системе AstraVer для доказательства условий корректности такие автоматические системы, как SMT-решатели, могут не справляться с моделированием работы с памятью в C-программах [44]. Применяемый нами подход, основанный на реализации сепарационной логики в системе VST, требует ручной работы при доказательстве корректности программ

с указателями, но при этом позволяет вручную приводить доказательство к успешному завершению.

Кроме того, в качестве родственного исследования рассмотрим проект по созданию системы C-lightVer для дедуктивной верификации C-программ [40, 41, 46]. Данная система позволяет упростить доказательство с помощью генерации вспомогательных лемм по определенным шаблонам [38, 39]. Но данная система, в отличие от применяемой нами системы VST, применима к ограниченному подмножеству языка программирования C [47, 48].

Родственным нашему проекту мероприятием является серия российских соревнований по формальной верификации программ VeHa. В рамках данной серии на текущий момент состоялись два соревнования, VeHa-2023 [16] и VeHa-2024 [10]. На данных соревнованиях команды исследователей также решают сложные для формальной верификации задачи. Но на нашем проекте, в отличие от соревнований серии VeHa, временные рамки намного более мягкие (две недели вместо трех дней), и, самое главное, на нашем проекте есть нацеленность на продолжение работы и после завершения самого мероприятия (Большой Математической Мастерской 2025 года).

Структура статьи Данная статья имеет следующую структуру: в главе 2 описаны применяемый нами метод дедуктивной верификации программ, применяемая нами система дедуктивной верификации VST и верифицируемая нами хэш-функция «Стрибог», в главе 3 описана проводимая нами дедуктивная верификация реализации хэш-функции «Стрибог», в заключении приведен список наших результатов и описаны наши планы на будущее, в Приложении A приведена предназначенная для упрощения доказательства корректности реализации XLPS лемма и схема ее доказательства.

Благодарности Авторы статьи выражают благодарность Группе Астра, а также лично Тимофею Юрьевичу Черганову, за запуск проекта «Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра»». Также авторы статьи выражают благодарность организаторам Большой Математической Мастерской 2025 года за организацию и поддержку проекта.

2. Основы дедуктивной верификации в системе VST и основы работы хэш-функции «Стрибог»

В данном разделе приводится описание используемых нами функций, методов и инструментария, а именно рассмотрены применяемый нами метод дедуктивной верификации программ, применяемая нами система дедуктивной верификации VST и верифицируемая нами хэш-функция «Стрибог».

2.1. Дедуктивная верификация программ

В 1969 году Ч. Хоар ввел способ задания аксиоматической семантики, ставший основой метода дедуктивной верификации программ. Подход Хоара заключается в том, чтобы представлять текст программы как особое отношение между утверждениями. Базовыми формулами в рассматриваемом подходе являются тройки Хоара $\{P\}\ S\ \{Q\}$, где P — предусловие (логическая формула), S — программа, Q — постусловие (логическая формула). Частичная корректность тройки Хоара $\{P\}\ S\ \{Q\}$ означает, что "если предусловие P истинно перед исполнением фрагмента программы S, и, если исполнение S завершилось, тогда постусловие Q выполняется после его завершения". Правила вывода задаются в виде

$$\frac{\psi_1, ..., \psi_n}{\varphi}$$

где ψ_1,...,ψ_n – посылки правила вывода (набор троек Хоара и логических формул) и φ — заключение правила вывода (тройка Хоара). Данная нотация означает, что φ выводимо при гипотезе ψ_1,...,ψ_n. Семантика простых операторов (например, присваивания) задается, как правило, с помощью набора аксиом, а любого сложного оператора (например, оператора последовательного исполнения) – с помощью правила вывода. Логическая система, содержащая аксиомы и правила вывода для всех синтаксических форм языка программирования, называется логикой Хоара или аксиоматической семантикой языка.

Генератор условий корректности осуществляет вывод в автоматическом режиме по аксиоматической семантике и сводит частичную корректность тройки Хоара $\{P\}\ S\ \{Q\}$ к истинности некоторого числа лемм, называемых условиями корректности, в предметной области. Доказуемости этих лемм достаточно для частичной корректности исходной аннотированной программы. Реализацией генератора условий корректности является система дедуктивной верификации. В нашем проекте в качестве системы дедуктивной верификации используется VST.

2.2. Система дедуктивной верификации VST

Рассмотрим схему дедуктивной верификации программ в системе VST, изображенную на рисунке 2.

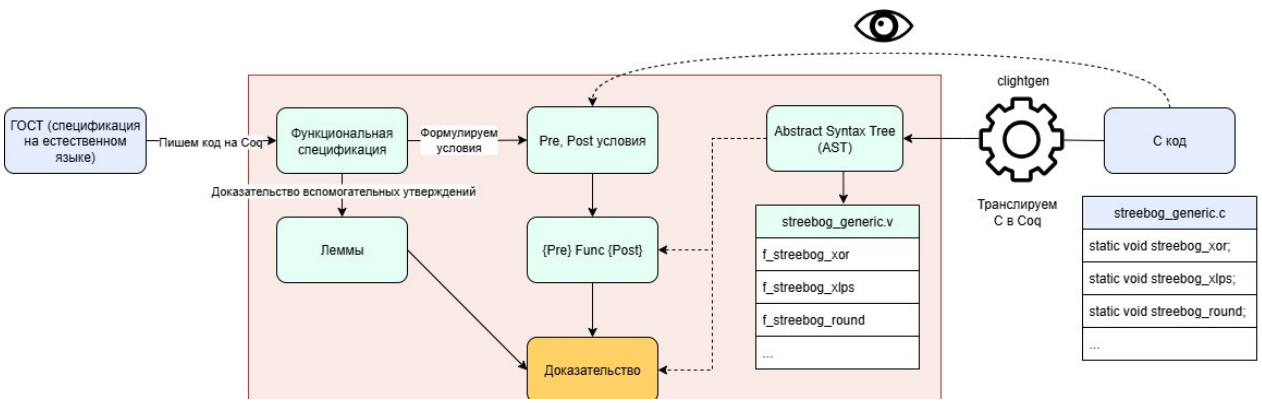


Рис. 2. Схема дедуктивной верификации программ в системе VST

Рассмотрим этапы верификации программ в системе VST в соответствии с данной схемой:

1. Задание спецификации программы в формальном виде на языке Gallina, входном языке системы доказательства Rocq. В случае нашего проекта функциональной спецификацией является ГОСТ Р 34.11-2012, то есть спецификация задана на естественном языке, и на данном этапе происходит задание стандарта ГОСТ Р 34.11-2012 в системе доказательства Rocq.
2. Задание для верифицируемых функций предусловий и постусловий. При этом для записи формул, выражающих предусловие и постусловие, активно используются заданные ранее определения из функциональной спецификации, а также имена переменных из C-программы.
3. Трансляция C-программы с помощью утилиты clightgen в абстрактное синтаксическое дерево (AST) в системе Rocq.
4. Связывание предусловий и постусловий с программой. Такое связывание происходит с помощью подключения модуля с AST-представлением программы, а затем заданием с помощью специальной функциональности VST леммы о корректности программы относительно предусловия и постусловия.
5. Задание дополнительных лемм для упрощения доказательства корректности программы и их доказательство. Такие леммы вместе с функциональной спецификацией формируют теорию предметной области.

6. Доказательство корректности программы относительно предусловия и постусловия. Такое доказательство происходит с помощью прямого прослеживания программы и доказательства полученных условий корректности прямо в момент их генерации при прямом прослеживании программы. Для упрощения доказательства возникающих таким образом условий корректности полезно использовать леммы из теории предметной области. Если при таком прямом прослеживании удается дойти до конца программы и доказать при этом все условия корректности, то программа корректна относительно предусловия и постусловия.

Отметим, что некоторые из данных этапов можно выполнять одновременно, например, задание спецификаций программ в формальном виде и трансляцию C-программы в AST-представление, или, например, задание лемм для упрощения доказательства корректности и само доказательство корректности. В нашем проекте данная схема применяется для дедуктивной верификации реализации хэш-функции «Стрибог» из ядра Linux.

2.3. Хэш-функция «Стрибог»

Рассмотрим схему работы хэш-функции «Стрибог», изображенную на рисунке 3.

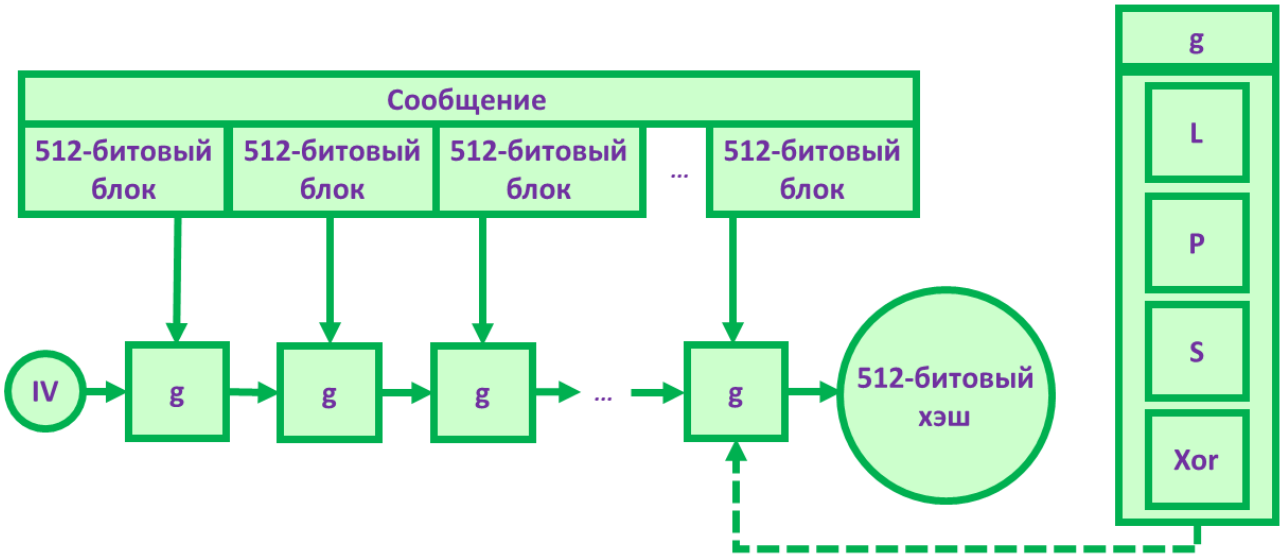


Рис. 3. Схема работы хэш-функции «Стрибог»

Данная хэш-функция принимает на вход инициализирующий вектор IV, а также сообщение, для которого нужно сгенерировать хэш. В качестве инициализирующего вектора в верифицируемом нами случае 512-битовой хэш-функции «Стрибог» используется 512-битовый нулевой вектор. Сообщение разделяется на части, каждая длиной 512 бит, а если

длина сообщения в битах не кратна 512, то последняя часть дополняется до 512-битового вектора нулями и одной единицей перед последней частью. Таким образом получается набор 512-битовых векторов. Хэш-функция «Стрибог» выполняет итерации над данным набором векторов, применяя на каждой итерации функцию сжатия g к текущей 512-битовой части сообщения и к результату предыдущей итерации. Функция сжатия основана на преобразовании XLPS, однако реализация данного преобразования в ядре Linux значительно отличается от описания из ГОСТ Р 34.11-2012 тем, что вместо описанных в стандарте преобразований (линейное преобразование L, подстановка P и нелинейное преобразование S) в рассматриваемой реализации используется преобразование со специальной матрицей. На первой итерации в качестве результата предыдущей итерации используется инициализирующий вектор. Результатом последней итерации является хэш всего сообщения.

3. Дедуктивная верификация реализации хэш-функции «Стрибог»

В данной главе мы презентуем такие результаты нашей работы, как формализацию ГОСТ Р 34.11-2012 в системе Rocq, задание для базовых функций из реализации хэш-функции «Стрибог» из ядра Linux предусловий и постусловий, доказательство корректности осуществляющей xor над 512-битовыми векторами функции "streebog_xor", расширение теории предметной области леммами об используемых в рассматриваемой реализации структурах данных и задание функциональной спецификации оптимизированной реализации преобразования XLPS из ядра Linux.

3.1. Формализация ГОСТ Р 34.11-2012 в системе интерактивного доказательства теорем Rocq

Так как ГОСТ Р 34.11-2012 задан в алгоритмическом стиле, то формализация стандарта в системе Rocq представляет собой кодирование алгоритмов из стандарта на языке Gallina. В результате получился файл "functional_spec.v", общедоступный в репозитории нашего проекта [55]. Отметим, что при задании алгоритмов из ГОСТ Р 34.11-2012 мы пытались придерживаться обозначений из стандарта, чтобы наша формализация была понятнее и другим исследователям, которые тоже могут использовать нашу формализацию в качестве спецификации. Также для упрощения понимания нашей формализации

другими исследователями мы по-возможности снабдили код комментариями.

ГОСТ Р 34.11-2012 основан на использовании таких типов данных, как 512-битовые вектора и разбиение 512-битовых векторов побайтово (на 8-битовые части). Для задания такого типа данных удобно иметь тип битового вектора параметрической длины, который можно в нашем случае параметризовать длиной 512, и длиной 8. Такой подход более гибок, чем подход из известной работы по формальной верификации SHA-256, где для представления 256-битовых векторов не использовались последовательности бит, а использовался список байтов. Мы тоже используем списки байтов для хранения разбиения 512-битовых векторов побайтово, однако мы кроме того используем тип данных для представления 512-битовых векторов в виде последовательности бит.

В качестве типа данных для представления последовательности бит параметрической длины мы использовали тип данных int из библиотеки compcert.lib.Integers. Данный тип представляет собой последовательность бит параметрической длины. Для данного типа уже предопределены параметризации по длине 8 (тип Byte), по длине 32 (тип Int) и по длине 64 (тип Int64). Тип Byte пригодился нам для хранения в виде списка байтов разбиение 512-битовых векторов на 8-битовые части. Для представления 512-битовых векторов мы параметризовали тип int длиной 512 и задали для такого типа в соответствии со стандартом названия block512 и Vec512.

Так как используемые в ГОСТ Р 34.11-2012 преобразования требуют конвертации из 512-битовых блоков в список байт и обратно, то мы задали функции для таких преобразований, которые названы block512_to_bytes и bytes_to_block512 соответственно. Вместе с другими вспомогательными функциями для данных типов мы сделали важные шаги на пути к задании теории для 512-битовых векторов и их разбиений на части, которая может быть использована для формальной верификации таких функций, где применяются 512-битовые вектора.

Использование библиотеки compcert.lib.Integers позволило нам использовать встроенные функции над машинными представлениями целых чисел. Отметим, что нам не пришлось задавить играющую важную роль в преобразовании XLPS функцию xor для 512-битовых векторов, так как после параметризации типа int мы автоматически получили функцию Vec512.xor для операции исключающего или над 512-битовыми векторами.

С помощью заданных нами типов данных мы определили алгоритмы из ГОСТ Р 34.11-2012 в виде функций на языке Gallina. Для базовых алгоритмов, выполняющих преобра-

зование LPS, мы задали функции с именами l, p и s соответственно. Для всего преобразования LPSX мы задали функцию LPSX как композицию функций l, p, s и Vec512.xor. Использование функция LPSX позволило нам определить функцию сжатия g под названием g_N. Итеративное исполнение хэш-функции «Стрибог», где применяется определенная нами функция сжатия g_N, реализована в функции stage2.

Отдельно отметим вопрос доверия к заданной нами формализации ГОСТ Р 34.11-2012. К сожалению, пока что нет индустриально применимых методов формальной верификации соответствия текста на естественном языке и кода в системе интерактивного доказательства. В стандарте предлагаются тесты, поэтому мы провели тестирование нашей формализации, реализовав применение всех тестов из стандарта в файле "functional_spec_test.v", доступном в репозитории нашего проекта [55]. Отметим, что в стандарте предлагаются тесты не только для запуска всей реализации хэш-функции «Стрибог», но и по-отдельности для базовых преобразований. Все предложенные в ГОСТ Р 34.11-2012 тесты были успешно пройдены нашей формализацией стандарта на языке Gallina. Заданная нами функциональная спецификация предназначена для использования в предусловиях и постусловиях реализаций хэш-функции «Стрибог», чтобы доказать эквивалентность таких реализаций нашей функциональной спецификации.

3.2. Задание предусловий и постусловий базовых функций из реализации хэш-функции «Стрибог» из ядра Linux

В первую очередь важно верифицировать базовые функции реализации хэш-функции «Стрибог» из ядра Linux, поэтому мы задали спецификации для следующих функций:

- "streebog_xor" – функция, осуществляющая операцию xor над 512-битовыми векторами, представленными в виде 8-ми 64-битовых частей. Предусловие и постусловие данной функции определены в конструкции streebog_xor_spec в файле "spec_streebog_xor.v", доступном в репозитории нашего проекта [55]. В постусловии данной функции мы задали ее эквивалентность функции Vec512.xor из функциональной спецификации.
- "streebog_init" – функция, осуществляющая инициализацию начального вектора. Предусловие и постусловие данной функции определены в конструкции streebog_init_spec в файле "spec_streebog_init.v", доступном в репозитории нашего проекта [55]. В постусловии данной функции мы задали, что в результате должен

получиться инициализирующий вектор. Также в целях дальнейшей верификации данной функции мы задали в конструкции memset_spec предусловие и постусловие применяемой в данной функции стандартной функции memset. Отметим, что задание спецификаций для функций стандартной библиотеки языка C является важной задачей.

- "streebog_add512" – функция, осуществляющая прибавление 512-битового вектора. Предусловие и постусловие данной функции определены в конструкции streebog_add512_spec в файле "spec_streebog_add512.v", доступном в репозитории нашего проекта [55]. В постусловии данной функции мы задали, что результате должна получиться сумма векторов.
- "streebog_xlps" – функция, осуществляющая преобразование XLPS. Предусловие и постусловие данной функции определены в конструкции streebog_xlps_spec в файле "spec_streebog_xlps.v", доступном в репозитории нашего проекта [55]. В постусловии данной функции мы задали ее эквивалентность функции lpsx из функциональной спецификации.

После задания предусловий и постусловий стало возможным доказательство корректности функций.

3.3. Доказательство корректности функции "streebog_xor", осуществляющей xor над 512-битовыми векторами

Нами была дедуктивно верифицирована функция "streebog_xor" осуществляющей xor над 512-битовыми векторами. Мы задали корректность данной функции относительно ее предусловия и постусловия в лемме body_streebog_xor и доказали данную лемму. Лемма body_streebog_xor и ее доказательство приведены в файле "spec_streebog_xor.v", доступном в репозитории нашего проекта [55].

В доказательстве леммы body_streebog_xor 24 раза применяется тактика forward, что позволяет осуществить прямое прослеживание по всем 24 инструкциям данной функции, внося факты о структуре программы в предусловие. Отметим, что данный процесс доказательство можно рассматривать, как доказательство эквивалентности функции "streebog_xor" и функции Vec512.xor из функциональной спецификации. Но структура функций "streebog_xor" и функции Vec512.xor существенно различается, что приводит к сложности такого доказательства. Функция Vec512.xor осуществляет xor над представле-

ниями векторов в виде последовательности бит, тогда как функция "streebog_xor" для эффективности исполнения осуществляет xor над представлениями векторов в виде 8-ми 64-битовых частей. Эти различия отражаются во внесенных в предусловие в ходе прямого прослеживания фактов о программе. В конце доказательства для вывода того, что из накопленного предусловия следует постусловие, применяется тактика entailer!, но перед этим для упрощения вывода нужно сформулировать и применить леммы о связи операции xor над 64-битовыми векторами и операции xor над последовательностями битов. Для этого потребовалось расширять теорию предметной области леммами об используемых в программе структурах данных (леммы xor_repr_comm, xor_unsigned_comm, Z_to_chunks_xor) и применять эти леммы.

3.4. Задание лемм о свойствах структур данных, используемых в реализации хэш-функции «Стрибог» из ядра Linux

Для доказательства корректности функция "streebog_xor" мы расширили теорию предметной области леммами о свойствах структур данных, применяемых в реализации хэш-функции «Стрибог» из ядра Linux. Данные леммы доступны в нашем репозитории [55] в файле "spec_streebog_xor.v", но так как свойства 64-битовых векторов могут пригодиться при верификации и других реализаций хэш-функции «Стрибог», то мы планируем перенести эти леммы в файл с функциональной спецификацией "functional_spec.v", чтобы сформировать теорию предметной области в одном файле.

Леммы xor_repr_comm, xor_unsigned_comm и Z_to_chunks_xor применяются непосредственно в доказательстве корректности функции "streebog_xor". Данные леммы описывают связь между операцией xor над последовательностями битов и операцией xor над битовыми векторами определенной длины. В качестве операции xor над последовательностями битов имеется в виду операция Z.lxor, определенная над типом Z, представляющим целое число в двоичном представлении с помощью беззнаковой или со знаком последовательности битов любой длины. Отметим, что для доказательства данных лемм были введены другие леммы, которые с одной стороны могут рассматриваться как вспомогательные, но с другой стороны тоже описывают важные свойства о связях двух видов операции xor.

Лемма xor_repr_comm описывает, что применение операции xor над 64-битовыми векторами к результатам приведения к 64-битному вектору двух последовательностей бит эк-

вивалентно приведению к типу 64-битового вектора результата применения Z.lxor к этим последовательностями бит. Приведем лемму xor_repr_comm и ее доказательство:

```
Lemma xor_repr_comm : forall x y,
  Int64.xor (Int64.repr x) (Int64.repr y) = Int64.repr (Z.lxor x y).
Proof.
  intros x y.
  specialize (Int64.same_bits_eq (Int64.xor (Int64.repr x) (Int64.repr y))
             (Int64.repr (Z.lxor x y))) as H; lapply H; clear H.
  - intros T; exact T.
  - intros i R. rewrite Int64.bits_xor.
    -- rewrite 3!Int64.testbit_repr.
       --- rewrite <- Z.lxor_spec.
           reflexivity.
       --- exact R.
       --- exact R.
       --- exact R.
    -- exact R.
Qed.
```

Доказательство данной леммы основано на применении лемм Int64.same_bits_eq и Int64.testbit_repr из библиотеки compcert.lib.Integers. Лемма Int64.same_bits_eq для 64-битовых векторов утверждает, что если все биты данных векторов равны, то и сами вектора равны. Лемма Int64.testbit_repr утверждает, что если индекс бита находится в диапазоне от 0 до длины машинного слова из 64 бит, то бит с таким индексом от результата приведения к 64-битовому вектору равен биту с тем же индексом от последовательности бит. Отметим, что определениях этих двух лемм основаны на операции Z.testbit, которая принимает на вход последовательность бит (имеет тип Z) и индекс бита и возвращает значение бита как булево значение.

Лемма xor_unsigned_comm описывает, что результат применения Z.lxor к беззнаковым приведениям к 512-битовым векторам двух последовательностей бит равен результату беззнакового приведения к 512-битовому вектору результата применения Z.lxor к этим последовательностям бит. Приведем лемму xor_unsigned_comm и ее доказательство:

```
Lemma xor_unsigned_comm : forall x y,
```

```
  Z.lxor (Vec512.unsigned x) (Vec512.unsigned y) =
  Vec512.unsigned (Vec512.xor x y).
Proof.
  intros [x Hx] [y Hy]. simpl.
  rewrite Vec512.Z_mod_modulus_eq.
  unfold Vec512.modulus in *.
  rewrite two_power_nat_equiv in *.
  rewrite Zlxor_mod_pow2 by easy.
  now rewrite 2!Zmod_small by lia.
Qed.
```

Доказательство данной леммы основано на применении заданной нами вспомогательной леммы Zlxor_mod_pow2. Данная лемма утверждает, что результат применения Z.lxor к двум аргументам, взятый по модулю степени двойки, равен результату применения Z.lxor к тем же аргументам, но взятым по модулю той же самой степени двойки. Приведем вспомогательную лемму Zlxor_mod_pow2 и ее доказательство:

```
Lemma Zlxor_mod_pow2 : forall x y n, 0 <= n ->
  (Z.lxor x y) mod 2 ^ n = Z.lxor (x mod 2 ^ n) (y mod 2 ^ n).
Proof.
  intros x y n Hn. apply Z.bits_inj'. intros i Hi.
  rewrite Z.lxor_spec. destruct (Z_lt_le_dec i n).
  - rewrite 3!Z.mod_pow2_bits_low by assumption.
    apply Z.lxor_spec.
  - now rewrite 3!Z.mod_pow2_bits_high.
Qed.
```

Доказательство данной вспомогательной леммы основано на рассмотрении двух случаев: случай для младших по отношению к степени двойки бит и случай для старших по отношению к степени двойки бит. Отметим, что в данной вспомогательной лемме формулируется относительно общее утверждение, которое может иметь обширные сферы применения при доказательстве теорем.

Лемма Z_to_chunks_xor описывает, что результат попарного применения Z.lxor к результатам разбиения последовательностей бит на одинаковое количество частей одинаковой длины равен результату попарного разбиения на то же количество частей той же

длины результата применения Z.lxor к этим последовательностям бит. Приведем лемму Z_to_chunks_xor и ее доказательство:

```
Lemma Z_to_chunks_xor : forall n m x y,
  map (uncurry Z.lxor) (combine (Z_to_chunks m n x) (Z_to_chunks m n y)) =
  Z_to_chunks m n (Z.lxor x y).
Proof.
   induction n; intros m x y; simpl.
  - reflexivity.
  - now rewrite <- xor_LSB_comm, Z.shiftr_lxor, IHn.
Qed.
```

Доказательство данной леммы основано на использовании заданной нами вспомогательной леммы xor_LSB_com. Данная вспомогательная лемма утверждает, что результат применения Z.lxor к j наименьшим значащим битам обоих аргументов равен j наименьшим значащим битам от результата применения Z.lxor к этим же аргументам. Приведем вспомогательную лемму xor_LSB_com и ее доказательство:

```
Lemma xor_LSB_comm : forall (j : nat) (x y : Z),
  Z.lxor (LSB j x) (LSB j y) = LSB j (Z.lxor x y).
Proof.
  intros j x y.
  unfold LSB.
  rewrite 3!Zbits.Z_mod_two_p_eq.
  rewrite <- Z.bits_inj_iff.
  unfold Z.eqf.
  intros n.
  rewrite Z.lxor_spec.
  specialize (Z.lt_ge_cases n (Z.of_nat j)) as [Hnltj | Hngej].
  - rewrite two_power_nat_equiv.
    rewrite 3!Z.mod_pow2_bits_low.
    --- rewrite <- Z.lxor_spec.
        reflexivity.
    --- exact Hnltj.
    --- exact Hnltj.
```

```
    --- exact Hnltj.
  - rewrite 3!testbit_ge_k.
    -- reflexivity.
    -- apply Z.le_ge. exact Hngej.
    -- apply Z.le_ge. exact Hngej.
    -- apply Z.le_ge. exact Hngej.
Qed.
```

Доказательство данной вспомогательной леммы основано на рассмотрении случаев для бит младше j-того и для бит старше j-того. Отметим, что в данной вспомогательной лемме также формулируется относительно общее утверждение, которое может упростить доказательство многих теорем.

Мы планируем пополнять теорию предметной области новыми леммами, которые будут упрощать доказательство корректности функций и описывать новые свойства типов данных, используемых при реализации хэш-функции «Стрибог». В качестве первоочередных задачи мы планируем доказать лемму о том, что описанная в ГОСТ Р 34.11-2012 композиция преобразований LPSX эквивалента применяемому в реализации хэш-функции «Стрибог» преобразованию со специальной матрицей.

3.5. Функциональная спецификация оптимизированной реализации преобразования XLPS из ядра Linux

Для доказательства корректности эффективной реализации преобразования XLPS из ядра Linux мы предложили использовать подход, состоящий из следующих шагов:

1. Задание функциональной спецификации оптимизированной реализации преобразования XLPS из ядра Linux.
2. Задание леммы об эквивалентности двух функциональных спецификаций XLPS: соответствующей стандарту и соответствующей эффективной реализации.
3. Применение такой леммы об эквивалентности для упрощения доказательства корректности реализации преобразования XLPS из ядра Linux.

Таким образом, в данном подходе для упрощения доказательства корректности оптимизированной программы вводится промежуточный этап в виде задания функциональной спецификации оптимизированной реализации и доказательства ее эквивалентности стандартной реализации.

Мы задали функциональную спецификацию оптимизированной реализации в файле "functional_spec.v" из нашего репозитория [55], дополнив тем самым теорию предметной области. Оптимизированный алгоритм реализует заданная нами на языке Gallina функция LPS_opt. Также мы задали интересующую нас лемму alg_equiv об эквивалентности LPS_opt и композиции l, p и s, но данная лемма пока что не доказана. Мы свели доказательство леммы alg_equiv к доказательству набора лемм, часть из которых еще предстоит доказать, и, таким образом, мы получили схему доказательства этой леммы. Лемма alg_equiv и схема ее доказательства приведены в Приложении А. Данная схема доказательства основана на материале статьи о том, как из стандартной реализации можно получить рассматриваемую оптимизированную реализацию [3].

4. Заключение

В данной статье мы презентуем следующие результаты:

- Формализацию ГОСТ Р 34.11-2012 в системе интерактивного доказательства Rocq. Данный результат может быть использован в качестве функциональной спецификации при верификации и других реализаций хэш-функции «Стрибог». Кроме того, данный результат демонстрирует актуальность формализации стандартов.
- Задание предусловий и постусловий базовых функций из реализации хэш-функции «Стрибог» из ядра Linux. Отметим, что была также заданы предусловие и постусловие стандартной функции memset. Данный результат позволяет верифицировать те функции, где применяются данные базовые функции. Кроме того, данный результат полезен не только для формальной верификации, но и для тестирования данных функций, позволяя проверять результат исполнения тестов с помощью проверки истинности постусловий по итогам тестов.
- Доказательство корректности функции xor над 512-битовыми векторами (функция "streebog_xor"), осуществляемая над данными векторами поблочно (8-блоков по 64 бита в каждом блоке). Данное доказательство позволяет доверять реализации данной функции. Поэтому, данную функцию можно использовать во всех программных системах, где требуется эффективная операция xor над 512-битовыми векторами.
- Задание лемм о свойствах структур данных, используемых в реализации хэш-функции «Стрибог» из ядра Linux. Данный результат позволяет упростить верификацию входящих в рассматриваемую реализацию функций с помощью переис-

пользования лемм при доказательстве корректности таких функций. Кроме того, данный результат может быть применим и при верификации других криптографических функций, так как в их реализациях могут встречаться подобные структуры данных.

- Задание функциональной спецификации оптимизированной реализации преобразования XLPS из ядра Linux. Данная функциональная спецификация ориентирована на упрощение доказательства корректности реализации XLPS из ядра Linux с помощью задания леммы об эквивалентности двух функциональных спецификаций XLPS: соответствующей стандарту и соответствующей оптимизированной реализации. Такое задание функциональных спецификаций оптимизированных программ, ориентированное на доказательство леммы об эквивалентности стандартной и оптимизированной версии программы, может быть примером подхода к дедуктивной верификации оптимизированных программ.

Отметим, что данные результаты могут служить прототипом комплексного подхода к дедуктивной верификации криптографических функций.

Мы рассматриваем планы по формальной верификации криптографических свойств реализации хэш-функции «Стрибог» из ядра Linux. Мы планируем завершить формальную верификацию реализации хэш-функции «Стрибог» из ядра Linux и сформировать по итогам такой верификации комплексный подход к верификации реализаций криптографических функций.

Список литературы

1. Васенин В.А., Кривчиков М.А. Формальные модели программ и языков программирования. Часть 1. Библиографический обзор 1930–1989 гг. // Программная инженерия. 2015. № 5. С. 10–19.
2. Васенин В.А., Кривчиков М.А. Формальные модели программ и языков программирования. Часть 2. Современное состояние исследований // Программная инженерия. 2015. № 6. С. 24–33.
3. Гафуров И.Р. Высокоскоростная программная реализация алгоритма хэширования «Стрибог» // Ученые записки УлГУ. Серия "Математика и информационные технологии". 2023. № 2. С. 19–27.
4. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования.
5. Девянин П.Н., Кулямин В.В., Петренко А.К., Хорошилов А.В., Щепетков И.В. Интеграция

мандатного и ролевого управления доступом и мандатного контроля целостности в верифицированной иерархической модели безопасности операционной системы // Труды Института системного программирования РАН. 2020. Т. 32. № 1. С. 7–26.

6. Девянин П.Н. , Леонова М.А. Приёмы описания модели управления доступом ОССН Astra Linux Special Edition на формализованном языке метода Event-B для обеспечения её верификации инструментами Rodin и ProB // Прикладная дискретная математика. 2021. № 52. С. 83–96.
7. Девянин П.Н., Тележников В.Ю., Хорошилов А.В. Формирование методологии разработки безопасного системного программного обеспечения на примере операционных систем // Труды Института системного программирования РАН. 2021. Т. 33. № 5. С. 25–40.
8. Камкин А.С. Введение в формальные методы верификации программ. М.: ДМК Пресс, 2024. – 304 с.
9. Кокорин А.О., Тиевский С.Д., Девянин П.Н. Приемы дедуктивной верификации программного кода с использованием AstraVer Toolset // Прикладная дискретная математика. Приложение. 2022. № 15. С. 80–90.
10. Кондратьев Д.А., Старолетов С.М., Шошмина И.В., Красненкова А.В., Зиборов К.В., Шилов Н.В., Гаранина Н.О., Черганов Т.Ю. Соревнования по формальной верификации VeHa-2024: накопленный в течение двух лет опыт и перспективы // Труды Института системного программирования РАН. 2025. Т. 37. № 1. С. 159–184.
11. Лейно К.Р.М. Доказательство корректности программ. М.: ДМК Пресс, 2024. – 522 с.
12. Миронов А.М. Методы верификации программ. М.: ДМК Пресс. 2023 – 332 с.
13. Непомнящий В.А., Рякин О.М. Прикладные методы верификации программ. М.: Радио и связь, 1988. – 256 с.
14. Операционная система Astra Linux. Режим доступа: `https://astralinux.ru` , свободный (дата обращения: 25.07.2025)
15. Седов Г.К. Стойкость ГОСТ Р 34.11-2012 к атаке поиска прообраза и к атаке поиска коллизий // Математические вопросы криптографии. 2015. Т. 6. № 2. С. 79–98.
16. Старолетов С.М., Кондратьев Д.А., Гаранина Н.О., Шошмина И.В. Соревнования по формальной верификации VeHa-2023: опыт проведения // Труды Института системного программирования РАН. 2024. Т. 36. № 2. С. 141–168.
17. Шилов Н.В. Основы синтаксиса, семантики, трансляции и верификации программ: учебное пособие. Новосибирск: Издательство Новосибирского государственного университета, 2011. – 292 с.
18. AlTawy R., Youssef A.M. Integral distinguishers for reduced-round Stribog // Information Processing Letters. 2014. Volume 114. Issue 8. pp. 426–431.
19. Appel A.W. Verification of a Cryptographic Primitive: SHA-256 // ACM Transactions on Programming Languages and Systems (TOPLAS). Volume 37. Issue 2. Article No.: 7. pp. 1–31.
20. Appel A.W. Verified Software Toolchain // Lecture Notes in Computer Science. 2011. Volume 6602. pp. 1–17.

21. Appel A.W., Beringer L., Cao Q., Dodds J. Verifiable C: Applying the Verified Software Toolchain to C programs. 2023. URL: `https://github.com/PrincetonUniversity/VST/raw/master/doc/VC.pdf` (Accessed 25 Jul 2025)
22. Apt K.R., Olderog E.-R. Assessing the Success and Impact of Hoare's Logic // Theories of Programming: The Life and Works of Tony Hoare. New York: ACM, 2021. pp. 41–76.
23. Apt K.R., Olderog E.-R. Fifty years of Hoare's logic // Formal Aspects of Computing. 2019. Volume 31. Issue 6. pp. 751–807.
24. Biryukov A., Perrin L., Udovenko A. Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1 // Lecture Notes in Computer Science. 2016. Volume 9665. pp. 372–402.
25. Brain M., Polgreen E. A Pyramid Of (Formal) Software Verification // Lecture Notes in Computer Science. 2025. Volume 14934. pp. 393–419.
26. Cao Q., Beringer L., Gruetter S., Dodds J., Appel A.W. VST-Floyd: A Separation Logic Tool to Verify Correctness of C Programs // Journal of Automated Reasoning. 2018. Volume 61. Issue 1. pp. 367–422.
27. Degtyarev A., Chikunov V. Streebog Hash Function. URL: `https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/crypto/streebog_generic.c` (Accessed 25 Jul 2025).
28. Dijkstra E.W., Schölten C.S. The strongest postcondition // Predicate Calculus and Program Semantics. New York: Springer, 1990. pp 209–215.
29. Efremov D., Mandrykin M., Khoroshilov A. Deductive verification of unmodified Linux kernel library functions // Lecture Notes in Computer Science. 2018. Volume 11245. pp. 216–234.
30. Filliâtre J.C. Deductive software verification // International Journal on Software Tools for Technology Transfer. 2011. Volume 13. Issue 5. Article ID: 397.
31. Floyd R.W. Assigning meanings to programs // Proc. Symposia in Applied Mathematics. Providence, 1967. Volume 19. pp. 19–32.
32. Guo J., Jean J., Leurent G., Peyrin T., Wang L. The Usage of Counter Revisited: Second-Preimage Attack on New Russian Standardized Hash Function // Lecture Notes in Computer Science. 2014. Volume 8781. pp. 195–211.
33. Hähnle R., Huisman M. Deductive Software Verification: From Pen-and-Paper Proofs to Industrial Tools // Lecture Notes in Computer Science. 2019. Volume 10000. pp. 345–373.
34. Harrison J. Theorem Proving for Verification (Invited Tutorial) // Lecture Notes in Computer Science. 2008. Volume 5123. pp. 11–18.
35. Hoare C.A.R. An axiomatic basis for computer programming // Communications of the ACM. 1969. Volume 12. Issue 10. pp. 576–580.
36. Ishtiaq S.S., O'Hearn P.W. BI as an assertion language for mutable data structures // Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL '01). London, United Kingdom, January 17–19, 2001. New York: Association for Computing Machinery, 2001. pp. 14–26.

37. Kazymyrov O., Kazymyrova V. Algebraic aspects of the Russian hash standard GOST R 34.11-2012 // Proceedings of the CTCrypt'13. Ekaterinburg, Russia, June 23–24, 2013. IACR ePrint, URL: `http://eprint.iacr.org/2013/556` (Accessed 25 Jul 2025)
38. Kondratyev D.A., Maryasov I.V., Nepomniaschy V.A. The Automation of C Program Verification by the Symbolic Method of Loop Invariant Elimination // Automatic Control and Computer Sciences. 2019. Volume 53. Issue 7. pp. 653–662.
39. Kondratyev D., Maryasov I., Nepomniaschy V. Towards Automatic Deductive Verification of C Programs over Linear Arrays // Lecture Notes in Computer Science. 2019. Volume 11964. pp. 232–242.
40. Kondratyev D.A., Nepomniaschy V.A. Automation of C Program Deductive Verification without Using Loop Invariants // Programming and Computer Software. 2022. Volume 48. Issue 5. pp. 331–346.
41. Kondratyev D.A., Promsky A.V. Developing a self-applicable verification system. Theory and practice // Automatic Control and Computer Sciences. 2015. Volume 49. Issue 7. pp. 445–452.
42. Ma B., Li B., Hao R., Li X. Improved Cryptanalysis on Reduced-Round GOST and Whirlpool Hash Function // Lecture Notes in Computer Science. 2014. Volume 8479. pp. 289–307.
43. Mandrykin M.U., Khoroshilov A.V. High-level memory model with low-level pointer cast support for Jessie intermediate language // Programming and Computer Software. 2015. Volume 41. Issue 4. pp. 197–207.
44. Mandrykin M.U., Khoroshilov A.V. Region analysis for deductive verification of C programs // Programming and Computer Software. 2016. Volume 42. Issue 5. pp. 257–278.
45. Mandrykin M.U., Khoroshilov A.V. Towards deductive verification of C programs with shared data // Programming and Computer Software. 2016. Volume 42. Issue 5. pp. 324–332.
46. Maryasov I.V., Nepomniaschy V.A., Promsky A.V., Kondratyev D.A. Automatic C Program Verification Based on Mixed Axiomatic Semantics // Automatic Control and Computer Sciences. 2014. Volume 48. Issue 7. pp. 407–414.
47. Nepomniaschy V.A., Anureev I.S., Mikhailov I.N., Promsky A.V. Towards verification of C programs. C-light language and its formal semantics // Programming and Computer Software. 2002. Volume 28. Issue 6. pp. 314–323.
48. Nepomniaschy V.A., Anureev I.S., Promskii A.V. Towards Verification of C Programs: Axiomatic Semantics of the C-kernel Languages // Programming and Computer Software. 2003. Volume 29. Issue 6. pp. 338–350.
49. O'Hearn P. Separation logic // Communications of the ACM. 2019. Volume 62. Issue 2. pp. 86–95.
50. O'Hearn P. Separation Logic Tutorial // Lecture Notes in Computer Science. 2008. Volume 5366. pp. 15–21.
51. Paulin-Mohring C. Introduction to the Coq Proof-Assistant for Practical Software Verification // Lecture Notes in Computer Science. 2012. Volume 7682. pp. 45–95.
52. Reynolds J.C. An Overview of Separation Logic // Lecture Notes in Computer Science. 2008. Volume 4171. pp. 460–469.

53. Reynolds J.C. Separation logic: a logic for shared mutable data structures // Proceedings 17th Annual IEEE Symposium on Logic in Computer Science. Copenhagen, Denmark, July 22–25, 2002. IEEE, 2002. pp. 55–74.
54. Saarinen M.-J. O. StriBob: authenticated encryption from GOST R 34.11-2012 LPS permutation // Mathematical Aspects of Cryptography. 2015. Volume 6. Issue 2. pp. 67–78.
55. Stribog Hash Formal Verification. URL: `https://github.com/MathAndLive/StribogHashFormalVerification` (Accessed 25 Jul 2025)
56. Volkov G., Mandrykin M. Efremov D. Lemma functions for Frama-C: C programs as proofs // Proceedings of the 2018 Ivannikov Ispras Open Conference (ISPRAS). Moscow, Russia, November 22–23, 2018. IEEE, 2018. pp. 31–38.
57. Wang Z., Yu H., Wang X. Cryptanalysis of GOST R hash function // Information Processing Letters. 2014. Volume 114. Issue 12. pp. 655–662.

A. Лемма об эквивалентности двух функциональных спецификаций XLPS: соответствующей стандарту и соответствующей оптимизированной реализации

Лемма alg_equiv задана следующим образом:

```
Lemma alg_equiv : forall (b : block512),
  LPS_opt b = l (p (s b)).
Proof.
  intros b.
  unfold LPS_opt.
  unfold tableLPS_i_j.
  assert ((fun k : Z => fold_right Int64.xor (Int64.repr 0)
  (map (fun i : Z => fold_right Int64.xor (Int64.repr 0)
  (map (fun k0 : Z => nthi_int64 A (63 - 8 * i - k0))
  (bit (Byte.unsigned
  (pi_byte (nthi_bytes (tau_bytes (block512_to_bytes b)) (8 * k + i)))))))
  [0; 1; 2; 3; 4; 5; 6; 7]))
          =
          the_thing b) as H by reflexivity; rewrite H; clear H.
  assert (map (the_thing b) [0; 1; 2; 3; 4; 5; 6; 7]
```

```
        =
        map (fun k : Z => fold_right Int64.xor (Int64.repr 0)
        (map (fun i : Z => fold_right Int64.xor (Int64.repr 0)
        (map (fun k0 : Z => nthi_int64 A (63 - 8 * i - k0))
        (bit (Byte.unsigned
        (nthi_bytes (block512_to_bytes (p (s b))) (8 * k + i))))))
        [0; 1; 2; 3; 4; 5; 6; 7]))
        [0; 1; 2; 3; 4; 5; 6; 7]) as H.
{
 unfold the_thing.
 pose proof map_ext_in as M.
 (* обязательное действие, без него specialize не сработает *)
 specialize (M
             Z
             int64
             (fun k : Z => fold_right Int64.xor (Int64.repr 0)
             (map (fun i : Z => fold_right Int64.xor (Int64.repr 0)
             (map (fun k0 : Z => nthi_int64 A (63 - 8 * i - k0))
             (bit (Byte.unsigned (pi_byte
             (nthi_bytes (tau_bytes (block512_to_bytes b)) (8 * k + i)))))))
             [0; 1; 2; 3; 4; 5; 6; 7]))
             (fun k : Z => fold_right Int64.xor (Int64.repr 0)
             (map (fun i : Z => fold_right Int64.xor (Int64.repr 0)
             (map (fun k0 : Z => nthi_int64 A (63 - 8 * i - k0))
             (bit (Byte.unsigned
             (nthi_bytes (block512_to_bytes (p (s b))) (8 * k + i))))))
             [0; 1; 2; 3; 4; 5; 6; 7]))
             [0; 1; 2; 3; 4; 5; 6; 7]).
 apply M.
 clear M.
 intros i R1.
 apply remove_fold_right_xor.
```

```
   pose proof map_ext_in as M. (* то же самое *)
   specialize (M
               Z
               int64
               (fun i0 : Z => fold_right Int64.xor (Int64.repr 0)
               (map (fun k0 : Z => nthi_int64 A (63 - 8 * i0 - k0))
               (bit (Byte.unsigned (pi_byte (nthi_bytes
               (tau_bytes (block512_to_bytes b)) (8 * i + i0)))))))
               (fun i0 : Z => fold_right Int64.xor (Int64.repr 0)
               (map (fun k0 : Z => nthi_int64 A (63 - 8 * i0 - k0))
               (bit (Byte.unsigned
               (nthi_bytes (block512_to_bytes (p (s b))) (8 * i + i0))))))
               [0; 1; 2; 3; 4; 5; 6; 7] ).
   apply M.
   clear M.
   intros l R2.
   apply remove_fold_right_xor.
   apply remove_map.
   apply f_equal.
   apply f_equal.
   apply ith_of_pi_tau.
   unfold In in R1.
   unfold In in R2.
   assert ((Datatypes.length tau) = 64%nat) as obvious by
   reflexivity; rewrite obvious; clear obvious.
   lia.
  }
  rewrite H; clear H.
  unfold l.
  apply f_equal.
  apply l_equiv.
Qed.
```

Схема доказательства данной леммы задана между конструкциями `Proof.` и `Qed`.

УДК 629.78:523.6

Геокосмический мониторинг околоземного пространства: перспективные технологии защиты от космического мусора и астероидной опасности

Колесенков А.Н., Гусев С.И., Аксенов А.А., Кондрашова Т.И., Пасичняк В.Е.
(ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина»)

Статья представляет комплексный обзор современных систем геокосмического мониторинга околоземного пространства, направленных на обеспечение защиты от космического мусора и астероидной опасности. В рамках обзора рассмотрены существующие и перспективные технологии, включая методы наблюдения, активного удаления угроз и применение информационных систем. Особое внимание уделено роли искусственного интеллекта и автоматизации в повышении эффективности мониторинга.

Ключевые слова: *геокосмический мониторинг, обзор, космический мусор, астероидная опасность, астероид, информационные технологии, спутниковые системы наблюдения, космическая безопасность.*

1. Введение

Современные вызовы требуют междисциплинарных решений, где информационные технологии играют ключевую роль в обработке данных и управлении системами мониторинга. Околоземное космическое пространство сегодня представляет собой динамически изменяющуюся среду, характеризующуюся растущим присутствием как естественных, так и искусственных объектов. Увеличение количества функционирующих космических аппаратов, рост популяции космического мусора и регулярное обнаружение новых потенциально опасных астероидов (ПОА) создают серьезные вызовы для обеспечения безопасности космической деятельности и защиты Земли от возможных катастрофических последствий столкновений [1].

Геокосмический мониторинг околоземного пространства как комплексная научно-техническая задача включает следующие основные направления [2]:

– обнаружение и каталогизацию объектов искусственного и естественного происхождения;

– определение и прогнозирование их орбитальных параметров;

– оценку рисков столкновений и потенциального ущерба;

– разработку и применение методов защиты от выявленных угроз.

По состоянию на начало 2025 года количество каталогизированных объектов космического мусора размером более 10 см превысило 34 000, в то время как число отслеживаемых потенциально опасных астероидов достигло 2 978. При этом наблюдается устойчивая тенденция к увеличению этих показателей, что обусловлено как совершенствованием средств наблюдения, так и продолжающимся техногенным загрязнением околоземного пространства [3].

Работа выполнена в формате аналитического обзора, систематизирующего последние достижения в области геокосмического мониторинга и защиты от космических угроз. Цель обзора — выделить ключевые технологические тренды, оценить их потенциал и обозначить барьеры внедрения. Материал основан на анализе научных публикаций, данных международных проектов (таких как ISON и «МАСТЕР») и открытых отчетов космических агентств.

2. Современное состояние околоземного космического пространства

Космический мусор включает нефункционирующие искусственные спутники, отработанные ступени ракет-носителей, фрагменты, образовавшиеся в результате столкновений и взрывов, а также технологические элементы, утерянные в процессе космической деятельности. Распределение этих объектов по орбитам и размерам представляет сложную картину, существенно влияющую на долгосрочную устойчивость космической деятельности [4].

Динамика роста количества каталогизированных объектов космического мусора за последние годы отражает как возрастающую активность в космосе, так и совершенствование средств наблюдения [5].

Особую обеспокоенность вызывает рост числа фрагментов, образующихся в результате столкновений и преднамеренных разрушений спутников в ходе испытаний противоспутникового оружия [6].

Потенциально опасными считаются астероиды, которые имеют минимальное расстояние пересечения орбиты с Землей (MOID) менее 0,05 а.е. (~7,5 млн км) и абсолютную

звездную величину H < 22 (что соответствует диаметру более 140 м). Столкновение с таким объектом может привести к региональной или глобальной катастрофе.

2.1. Системы мониторинга околоземного пространства

Основу глобальной системы мониторинга околоземного пространства составляют наземные средства наблюдения, включающие оптические телескопы и радиолокационные станции [7].

Космические системы мониторинга обладают рядом преимуществ по сравнению с наземными средствами, включая отсутствие атмосферных искажений, возможность непрерывного наблюдения и более выгодное геометрическое положение относительно наблюдаемых объектов [8].

Эффективный мониторинг околоземного пространства требует интеграции различных систем наблюдения и обмена данными между ними. В настоящее время существует несколько международных инициатив, направленных на координацию усилий в этой области.

2.2 Международные инициативы по координации мониторинга

Эффективный мониторинг околоземного пространства требует глобальной кооперации. Ключевые действующие инициативы включают:

Проект ISON (International Scientific Optical Network)

Крупнейшая международная сеть оптических обсерваторий, объединяющая 23 обсерватории в Европе, Азии, Африке и Америке.

Основные задачи:

– Трекер 10 000+ объектов (GEO, HEO, LEO), включая космический мусор и астероиды.

– Автоматизированное управление телескопами через ПО (например, FORTE) и обработка данных с использованием ИИ.

Сеть обсерваторий БРИКС

Развивается с 2019 года как часть сотрудничества стран БРИКС.

Пример: российско-китайские наблюдения фрагментов ступени «Центавр» — первый шаг к созданию единой сети [14].

Цели: совместные кампании по отслеживанию мусора и опасных астероидов, стандартизация данных.

Программа ESA

ESA’s Space Debris Office: координация данных от радиолокаторов и телескопов, включая сотрудничество с ISON.

4. Методы защиты от космического мусора и астероидной опасности

Пассивные методы защиты направлены на минимизацию вероятности образования нового космического мусора и снижение рисков столкновений с существующими объектами.

Активные методы защиты включают в себя системы уклонения от столкновений и технологии активного удаления космического мусора.

Эффективность маневров уклонения зависит от точности определения орбит, своевременности обнаружения угрозы и маневренных возможностей космического аппарата [9].

Основной стратегией защиты от астероидной опасности является раннее обнаружение потенциально опасных объектов и определение их орбитальных и физических характеристик.

Для предотвращения столкновения с Землей потенциально опасного астероида необходимо изменить его траекторию таким образом, чтобы обеспечить пролет на безопасном расстоянии [10].

5. Информационные технологии в системах защиты от космического мусора и астероидной опасности

Современные системы геокосмического мониторинга и защиты от угроз невозможны без применения передовых информационных технологий (ИТ). Они обеспечивают сбор, обработку, анализ и распределение данных, а также автоматизированное управление наблюдательными комплексами.

5.1. Программное обеспечение для управления сетями телескопов

Ключевую роль играют специализированные программные комплексы, координирующие работу распределённых обсерваторий. Среди наиболее успешных примеров:

Проект ISON

Использует автоматизированную систему управления роботизированными оптическими обсерваториями, которая включает:

– планирование наблюдений с учётом погодных условий и приоритетов задач;

– обработку изображений в реальном времени с применением алгоритмов машинного обучения для идентификации объектов;

– интеграцию с базами данных орбитальных параметров для оперативного обновления каталогов [15].

Глобальная сеть «МАСТЕР»

Основана на концепции мультифункционального астрономического комплекса с динамически интегрированной базой данных. Её ПО обеспечивает:

– многоканальные наблюдения (оптические, УФ, радио);

– автоматическое распознавание транзиентных событий (включая потенциально опасные астероиды);

– координацию работы телескопов в разных часовых поясах для непрерывного мониторинга [16].

Под транзиентными событиями в мониторинге околоземного пространства понимаются кратковременные или изменяющиеся явления, такие как:

– вспышки на поверхности объектов (например, отражение солнечного света от вращающегося астероида);

– внезапное появление новых объектов (фрагменты мусора после столкновений, ранее не каталогизированные астероиды);

– аномальные изменения траекторий (резкие маневры спутников, гравитационные возмущения).

5.2. Искусственный интеллект и Big Data в мониторинге

Алгоритмы машинного обучения [17] применяются для:

– Классификации объектов (мусор/астероид/спутник) на основе спектральных и орбитальных характеристик.

– Прогнозирования столкновений с учётом неопределённостей орбит (методы Monte Carlo, нейросетевые модели).

Обработка больших данных [18]:

– Агрегация информации от радиолокаторов, лидаров и телескопов в единые облачные платформы (например, ESA’s Space Debris Office).

– Использование графовых баз данных для анализа связей между объектами и фрагментами после столкновений.

5.3. Моделирование и цифровые двойники

Физические симуляторы (например, NASA’s ADAM) воспроизводят:

– Динамику облаков мусора после фрагментации.

– Эффективность методов удаления (лазерная абляция, тросовые системы).

– Цифровые двойники астероидов помогают тестировать стратегии отклонения, учитывая состав и структуру объекта.

6. Перспективные технологии

Развитие технологий мониторинга околоземного пространства (табл. 1) идет по пути повышения чувствительности датчиков, расширения зоны наблюдения и автоматизации процессов обнаружения и сопровождения объектов [11].

Некоторые методы требуют уточнения терминологии:

Абляция (от лат. ablatio — «удаление») — процесс удаления вещества с поверхности объекта под воздействием внешних факторов (лазерного излучения, трения об атмосферу и др.). В контексте борьбы с космическим мусором и астероидами используется для изменения орбиты объекта за счёт импульса от испарения его материала.

Импактор (от англ. impact — «удар») — аппарат, предназначенный для столкновения с объектом (например, астероидом) с целью передачи кинетической энергии и изменения его траектории.

Таблица 1. Перспективные технологии мониторинга околоземного пространства

Технология	**Принцип действия**	**Потенциальные преимущества**	**Временной горизонт внедрения**
ИТ-платформы для распределённого мониторинга	Агрегация данных с глобальных сетей телескопов и радаров	Высокая скорость обработки, минимизация «слепых» зон	2-3 года
Многопозиционные радиолокационные системы	Использование разнесенных передатчиков и приемников	Повышение чувствительности, снижение энергопотребления	3-5 лет
Квантовые датчики	Использование квантовых эффектов для повышения	Обнаружение более мелких объектов	10-15 лет

	чувствительности		
Космические лидары	Активное лазерное зондирование из космоса	Высокая точность определения расстояния и скорости	5-8 лет
Сверхширокополосные радары	Использование сигналов с очень широким спектром	Высокое разрешение по дальности	5-7 лет
Распределенные группировки малых спутников	Координированная работа множества малых КА	Непрерывное глобальное покрытие, устойчивость к отказам	3-6 лет
Адаптивная оптика космического базирования	Коррекция искажений в режиме реального времени	Повышение разрешающей способности оптических систем	7-10 лет
Гиперспектральное зондирование	Анализ излучения в сотнях спектральных каналов	Определение состава и структуры объектов	4-7 лет
Искусственный интеллект для автоматического обнаружения	Применение методов машинного обучения	Повышение эффективности обнаружения, снижение ложных тревог	2-4 года

Развитие технологий защиты от космического мусора (табл. 2) направлено на повышение эффективности удаления существующих объектов и предотвращение образования нового мусора [12].

Таблица 2. Перспективные технологии активного удаления космического мусора

Технология	**Принцип действия**	**Потенциальные преимущества**	**Временной горизонт внедрения**
Электродинамические тросовые системы	Использование взаимодействия	Не требует расхода рабочего тела,	5-8 лет

	проводящего троса с магнитным полем Земли	эффективно для низких орбит	
Роботизированные системы с искусственным интеллектом	Автономные роботы с продвинутыми алгоритмами распознавания и захвата	Повышение эффективности захвата некооперативных объектов	7-10 лет
Многоразовые буксиры на электрореактивных двигателях	КА многократного использования для последовательного увода нескольких объектов	Экономическая эффективность	5-8 лет
Лазерные системы наземного базирования	Мощные лазеры для абляции поверхности объектов	Возможность воздействия на множество мелких фрагментов	8-12 лет
Пена для замедления объектов	Распыление специальной пены, увеличивающей аэродинамическое сопротивление	Пассивный метод, не требующий точного наведения	10-15 лет
Самовосстанавливающиеся защитные экраны	Материалы, способные "залечивать" повреждения от ударов	Повышение долговечности КА в условиях загрязненных орбит	10-15 лет
Универсальный интерфейс для будущего обслуживания	Стандартизированные точки захвата и стыковки на всех КА	Упрощение будущего обслуживания и удаления	3-5 лет

Развитие технологий защиты от астероидной опасности (табл. 3) направлено на повышение эффективности воздействия на астероиды различных размеров и составов [13].

Таблица 3. Перспективные технологии защиты от астероидной опасности

Технология	**Принцип действия**	**Потенциальные преимущества**	**Временной горизонт внедрения**

Ядерные взрывные устройства направленного действия	Фокусировка энергии взрыва в заданном направлении	Повышение эффективности передачи импульса	10-15 лет
Высокоскоростные кинетические импакторы (>30 км/с)	Использование гравитационных маневров и передовых двигательных технологий	Значительное увеличение передаваемого импульса	7-10 лет
Лазерные системы космического базирования	Высокомощные лазеры для абляции поверхности астероида	Возможность длительного контролируемого воздействия	15-20 лет
Гравитационные тракторы с дополнительным воздействием	Комбинация гравитационного воздействия с ионными двигателями, направленными на астероид	Повышение эффективности гравитационного трактора	10-15 лет
Проникающие устройства с термоядерными зарядами	Устройства, способные проникать вглубь астероида перед детонацией	Повышение эффективности передачи энергии	15-20 лет
Роевые системы воздействия	Множество малых КА, координированно воздействующих на астероид	Адаптивность, устойчивость к отказам	15-20 лет

7. Заключение

Проведенный обзор демонстрирует, что геокосмический мониторинг околоземного пространства и разработка методов защиты от космического мусора и астероидной опасности представляют собой комплексную научно-техническую проблему, решение которой требует междисциплинарного подхода и международного сотрудничества. Развитие информационных технологий — от автоматизированных систем управления телескопами (ISON, МАСТЕР) до AI-аналитики Big Data — становится критическим фактором в повышении точности и оперативности реагирования на космические угрозы. Дальнейшая

интеграция информационных технологий с аппаратными комплексами, включая квантовые вычисления для ускорения орбитальных расчётов, определит конкурентоспособность защитных систем в ближайшее десятилетие.

Проведенный в статье анализ показывает, что:

– Современное состояние околоземного пространства характеризуется устойчивым ростом популяции космического мусора, что создает возрастающие риски для функционирующих космических аппаратов. При этом наблюдается увеличение темпов запусков новых спутников, особенно в составе крупных группировок.

– Системы мониторинга космического мусора и астероидной опасности активно развиваются, но все еще имеют ограничения по обнаружению и каталогизации объектов малых размеров. Полнота каталогизации космического мусора размером менее 10 см на низких орбитах и менее 1 м на геостационарной орбите остается недостаточной.

– Потенциально опасные астероиды размером более 140 м обнаружены примерно на 56%, что оставляет значительную неопределенность в оценке астероидной угрозы. При этом для объектов размером менее 100 м полнота обнаружения составляет менее 30%.

– Технологии активной защиты от космического мусора находятся на стадии демонстрационных миссий и пока не применяются в промышленных масштабах. Наиболее перспективными представляются технологии контактного захвата для крупных объектов и бесконтактные методы для множественного воздействия на мелкие фрагменты.

– Методы защиты от астероидной опасности прошли первую успешную демонстрацию, но требуют дальнейшего развития для обеспечения эффективности при различных сценариях угрозы. Комбинированные подходы, включающие кинетическое, гравитационное и взрывное воздействие, представляются наиболее перспективными.

– Международное сотрудничество в области мониторинга и защиты от космических угроз расширяется, но сохраняются значительные политические и организационные барьеры. Требуется совершенствование международно-правовой базы и механизмов координации усилий.

– Перспективные технологии мониторинга и защиты могут существенно повысить эффективность противодействия космическим угрозам, но требуют значительных инвестиций и международной кооперации для своевременной разработки и внедрения.

Список литературы

1. Мануйлов С. А. Космический мусор - угроза безопасности космических полетов / С. А. Мануйлов // Проблемы безопасности полетов. – 2021. – № 9. – С. 35-53.

2. Молотов И. Е. Проблемы оптического мониторинга космического мусора /, В. М. Агапов, А. И. Стрельцов [и др.] // Препринты ИПМ им. М.В. Келдыша. – 2020. – № 7. – С. 1-17.
3. Ипатов А. В. Мониторинг космического мусора и перспективы освоения околоземного космического пространства / А. В. Ипатов, Л. А. Ведешин, С. А. Герасютин // Земля и Вселенная. – 2024. – № 2(356). – С. 91-108.
4. Ваганов А. Г. Защита от астероидной опасности: возможные варианты / А. Г. Ваганов // Энергия: экономика, техника, экология. – 2017. – № 8. – С. 75-80.
5. Усовик И. В. Разработка системы мониторинга некаталогизируемого космичесого мусора / И. В. Усовик, А. А. Морозов // Вестник Московского авиационного института. – 2023. – Т. 30, № 4. – С. 202-209.
6. Тесленко Е. С. Космический мусор как риск безопасности космической деятельности / Е. С. Тесленко // XXXVI Международные Плехановские чтения : Сборник статей участников конференции. В 4-х томах, Москва, 25–27 апреля 2023 года. – Москва: Российский экономический университет им. Г.В. Плеханова, 2023. – С. 14-19.
7. Лошкарева Е. Важность развития Системы Контроля Космического Пространства России и Европы для борьбы с космическим мусором / Е. Лошкарева, Т. Б. Радченко // Секция "Актуальные проблемы социально-гуманитарного знания" : Сборник докладов в рамках Московской молодёжной научно-практической конференции, Москва, 21–23 апреля 2015 года / Ответственный редактор: Каллиопин А. К.. – Москва: Издательство "Перо", 2015. – С. 62-71.
8. Сирый Р. С. Возможности применения орбитальных систем компьютерного зрения для мониторинга космического мусора / Р. С. Сирый, П. С. Баранов // Фундаментальные и прикладные космические исследования : XX Конференция молодых учёных, Москва, 15–14 апреля 2023 года. – Москва: Институт космических исследований Российской академии наук, 2023. – С. 139-150.
9. Палюх Б. В. Методика выбора способа обеспечения безопасности космического аппарата при возможном воздействии объектов космического мусора / Б. В. Палюх, В. К. Кемайкин, И. В. Кожухин // Мягкие измерения и вычисления. – 2018. – № 8(9). – С. 35-44.
10. Николаева, Е. А. Преодоление астероидной опасности. Методы противодействия астероидной опасности / Е. А. Николаева // 14-я Международная конференция "Авиация и космонавтика - 2015" Тезисы, Москва, 16–20 ноября 2015 года / Московский авиационный институт (национальный исследовательский университет). – Москва: Московский авиационный институт (национальный исследовательский университет), 2015. – С. 70-72.
11. Камысбаева С. М. Анализ влияния космического мусора на космические объекты / С. М. Камысбаева // Научное сообщество студентов XXI столетия. Технические науки : сборник статей по материалам CXLIV студенческой международной научно-практической конференции, Новосибирск, 05 декабря 2024 года. – Новосибирск: Общество с ограниченной ответственностью "Сибирская академическая книга", 2024. – С. 73-77.

12. Кулагин В. П. Проблемы астероидной опасности. Современные технологии и способы решения /, А. Ф. Каперко, А. А. Ледков, Б. М. Шустов // Новые информационные технологии : Тезисы докладов XXI Международной студенческой школы-семинара, Судак, 20–27 мая 2013 года / МИЭМ НИУ ВШЭ. – Судак: Московский институт электроники и математики НИУ ВШЭ, 2013. – С. 38-40.
13. Орлов П. Ю. Космический мусор как угроза для освоения околоземного космического пространства / П. Ю. Орлов, А. В. Панкин // Конструкторское бюро. – 2017. – № 3. – С. 62-68.
14. Molotov, I. & Sun, R. & Zhang, Chen & Almeida, U. & Zakhvatkin, Mikhail & Titenko, V. & Streltsov, A. & Mokhnatkin, A. & Kokina, T. & Elenin, Leonid. (2019). Russian-Chinese Observations of Fragments of the Destruction of the Centaur Rocket Stage are the First Step to the Network of BRICS Observatories. 95-102. 10.21046/spacedebris2019-95-102.
15. Elenin, L.V., Molotov, I.E. Software for the Automated Control of Robotic Optical Observatories. J. Comput. Syst. Sci. Int. 59, 894–904 (2020). https://doi.org/10.1134/S1064230720040036.
16. Lipunov, V.M., Vladimirov, V.V., Gorbovskoi, E.S. et al. The Concept of a Multi-Functional Astronomy Complex and Dynamically Integrated Database Applied to Multi-Channel Observations with the MASTER Global Network. Astron. Rep. 63, 293–309 (2019). https://doi.org/10.1134/S1063772919040073.
17. Jharbade, P., & Dixit, M. (2022). Detecting Space Debris using Deep Learning Algorithms: A Survey. 883–890. https://doi.org/10.1109/ICIRCA54612.2022.9985622.
18. Erik Blasch, Mark Pugh, Carolyn Sheaff, Joe Raquepas, Peter Rocci, "Big data for space situation awareness," Proc. SPIE 10196, Sensors and Systems for Space Applications X, 1019607 (5 May 2017); https://doi.org/10.1117/12.2264684.

УДК 519.681.2

Процессный подход к верификации криптографических протоколов*

Миронов А.М. (Московский Государственный Университет имени М.В. Ломоносова)

В настоящей работе излагается новая математическая модель криптографических протоколов, и приводятся примеры применения этой модели для решения задач верификации криптографических протоколов. Криптографические протоколы – это коммуникационные протоколы, реализованные с применением криптографических алгоритмов для решения задач защиты информации, в рамках которого стороны информационного взаимодействия последовательно выполняют определенные действия и обмениваются сообщениями. Они используются, например, в электронных платежах, электронных процедурах голосования, системах доступа к конфиденциальным данным, и т.д. Ошибки в криптографических протоколах могут привести к большому ущербу, поэтому необходимо использовать математические методы для обоснования различных свойств корректности и безопасности криптографических протоколов. В работе излагаются новые методы формальной верификации криптографических протоколов. Для моделирования криптографических протоколов в работе вводятся понятия последовательного и распределенного процессов. Предлагаемый подход предназначен только для доказательства корректности криптографических протоколов. Особенностью модели протоколов является её простота по сравнению с другими моделями протоколов, основанных на логических формулах или на алгебраических процессных выражениях. Участники протоколов представляются в виде графов, представляющих системы переходов. Действия, выполняемые участниками, являются метками этих переходов. Методы обоснования корректности протоколов, рассматриваемые в настоящей статье, связаны с рассуждениями для графов, которые более просты и наглядны по сравнению с методами, основанными на построении логического вывода в логических и алгебраических моделях протоколов.

Ключевые слова: криптографические протоколы, последовательные процессы, распределенные процессы, верификация

*при поддержке Математического центра в Академгородке, соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2025-349

1. Введение

Представленное исследование является развитием исследования, описанного в [2].

1.1. Понятие криптографического протокола

Криптографический протокол (КП) – это коммуникационный протокол, реализованный с применением криптографических алгоритмов для решения задач защиты информации, в рамках которого стороны информационного взаимодействия последовательно выполняют определенные действия и обмениваются сообщениями. КП представляет собой распределенный алгоритм, описывающий порядок обмена сообщениями между несколькими агентами. Примеры таких агентов – компьютерные системы, банковские карточки, люди, и т.д.

Для обеспечения свойств безопасности КП (таких например как конфиденциальность передаваемых данных) в КП могут использоваться криптографические преобразования (шифрование, электронная подпись, хэш-функции, и т.п.). Мы предполагаем, что криптографические преобразования, используемые в КП, являются идеальными, т.е. удовлетворяют некоторым аксиомам, выражающим, например, невозможность извлечения открытых текстов из шифртекстов без знания соответствующих криптографических ключей.

1.2. Уязвимости в криптографических протоколах

Многие уязвимости в КП связаны не с плохими криптографическими качествами используемых в них криптографических примитивов, а с логическими ошибками в КП. Наиболее ярким примером уязвимости в КП является уязвимость в КП аутентификации Нидхэма-Шредера [122], который был опубликован в 1978 г., и использовался в критических по безопасности информационных системах. Спустя более 16 лет после начала использования этого КП в нем обнаружилась логическая ошибка [111], связанная с возможностью непредусмотренного нечестного поведения одного из участников этого КП и нарушающая свойство безопасности этого КП. Особенность этой ошибки заключается в том, что данный КП является предельно простым распределенным алгоритмом, состоящим всего из трех действий, и при визуальном анализе этого КП отсутствие в нем ошибок не вызывало никаких сомнений. Ошибка была обнаружена лишь при помощи инструмента автоматизированной верификации КП.

Другой пример логической ошибки в КП (взят из статьи [70]): в КП входа в портал

Google, позволяющем пользователю идентифицировать себя только один раз, а затем обращаться к различным приложениям (таким, например, как Gmail или календарь Google), обнаружена логическая ошибка, позволяющая нечестному поставщику услуг выдавать себя за любого из своих пользователей для другого поставщика услуг.

Существует много других примеров КП (см. например [120], [118], [110], [78]), в которых обнаружились уязвимости следующего вида:

- участники этих КП могут получать искаженные сообщения (или вообще терять их) в результате перехвата, удаления или искажения противником передаваемых сообщений, что нарушает свойство целостности передаваемых сообщений,
- противник может узнать секретную информацию, содержащуюся в перехваченных сообщениях, в результате чего нарушается свойство конфиденциальности передаваемых сообщений.

Также есть примеры уязвимостей в КП, используемых для аутентификации перед провайдерами мобильной телефонной связи, для снятия денег в банкомате, для работы с электронными паспортами, проведения электронных выборов, и т.д.

Все эти примеры являются обоснованием того, что в критических по безопасности системах недостаточно неформального анализа требуемых свойств безопасности используемых в них КП, необходимо

- построение **математических моделей** анализируемых КП,
- описание свойств анализируемых КП в виде математических объектов, называемых **спецификациями** свойств этих КП, и
- построение формальных доказательств утверждений о том, что анализируемые КП удовлетворяют (или не удовлетворяют) своим спецификациям, процедура построения таких доказательств называется **верификацией** анализируемых КП.

В настоящей работе строится новая математическая модель КП, в терминах которой можно выражать такие свойства корректности КП, как например конфиденциальность передаваемых сообщений (т.е. обоснование следующего свойства анализируемого КП: содержание сообщений, посланных одним участником этого КП другому участнику этого КП, не будет известно противнику), или аутентификация (т.е. доказательство подлинности) участников КП.

1.3. Основные методы моделирования и верификации криптографических протоколов

Обзоры наиболее широко используемых методов моделирования и верификации КП содержатся в книгах [74] и [72]. Основные классы моделей КП и подходов к верификации КП имеют следующий вид.

1. **Логические модели**.

   Данный класс моделей был самым первым подходом к моделированию и верификации КП. На основе данного класса моделей проблема верификации КП сводится к проблеме построения в некотором логическом исчислении доказательства теоремы о том, что анализируемый КП обладает заданными свойствами. В работе [116] была изложена первая математическая модель КП, называемая **логикой BAN** (название этой логики соответствует фамилиям ее создателей – Бэрроуза, Абади и Нидхэма). Данная модель имеет большие ограничения: в ней предполагается, что участники анализируемого КП являются честными, т.е. точно выполняют предписания КП. Такое ограничение не позволило обнаружить упомянутую выше уязвимость в КП Нидхэма-Шредера. Кроме того, данная модель не позволяет анализировать КП с неограниченным порождением сеансов. Аппарат логики BAN был развит в работах [117], [115], [113], [114], [112], [108], [88]. Важным классом логических исчислений для моделирования и анализа КП является композиционная логика протоколов (Protocol Composition Logic), которой посвящены работы [91], [80], [77], [75]. Одним из классов логических моделей КП связан с логическим программированием. В данных моделях шаги протокола представляются в виде правил переписывания термов. Для моделирования КП используются клаузы Хорна и системы уравнений с ограничениями (constraint systems). Данный подход излагается в работах [90], [84] и др.

   Важным классом логических методов моделирования и анализа КП является индуктивный метод Паульсона: [106], [100], [97], [92].

2. Модели, основанные на **алгебре процессов**.

   Источником данного класса моделей является основополагающая работа Р.Милнера [121]. В данной работе строится модель взаимодействующих процессов, в которой процессы представляются термами. На этих термах вводится отношение наблюдаемой эквивалентности, которое позволяет эффективно выражать различные свойства

процессов, связанные с безопасностью (в частности свойства секретности и анонимности). Первой работой, в которой излагается модель КП на базе подхода Р.Милнера, является статья М.Абади и А.Гордона [96]. Среди других работ, относящихся к этому направлению, можно отметить работы [95], [89], [86], [79], [76], [67], [68], [66], [64].

3. Модели, основанные на **CSP**.

   CSP (**Communicating Sequential Processes**) – это математический аппарат, разработанный Ч.Хоаром [119] и предназначенный для моделирования и анализа распределенных вычислительных процессов. На базе этого аппарата построен метод моделирования и верификации КП, наиболее полно изложенный в книге [94]. Дедуктивная верификация КП на основе данного подхода использует понятие **ранг-функции**. Среди работ, относящихся к данному направлению, можно отметить работы [109], [107], [105], [104], [103], [65].

4. Модели, основанные на **пространствах нитей** (**strand spaces**).

   Пространства нитей позволяют представлять процессы, входящие в КП, в виде графических объектов (называемых нитями), в которых указаны зависимости между действиями, относящимися к различным процессам. Среди работ, относящихся к методам моделирования и верификации КП на основе понятия пространства нитей, можно отметить работы [101], [102], [98], [99], [93], [87], [85], [81], [82], [83], [73], [71], [69].

1.4. Сравнение предлагаемой модели криптографических протоколов с другими моделями

Модель КП, излагаемая в настоящей работе, унаследовала наиболее существенные качества моделей каждого из перечисленных выше четырех классов. В этой модели КП представляются в виде распределенных процессов (РП), взаимодействующих путем асинхронной передачи сообщений через каналы. Каждый РП, соответствующий какому-либо КП, представляет собой совокупность последовательных процессов (ПП), моделирующих работу участников этого КП. Как правило,

- эти ПП представляют собой последовательности действий, которые графически можно изобразить в виде нитей, и
- выполнение всего КП можно представить в виде пространства нитей, точки на которых связаны ребрами, изображающими передачу и прием сообщений.

Свойства КП могут представляться в виде логических формул, для обоснования которых могут использоваться стандартные алгоритмы логического вывода. Кроме того, некоторые свойства КП (например анонимность) м.б. выражены в виде отношения наблюдаемой эквивалентности между соответствующими РП, аналогично тому, как это делается в моделях КП основанных на процессной алгебре.

Основное достоинство предложенной модели заключаются в том, что доказательства свойств корректности КП на основе данной модели м.б. существенно короче, чем доказательства этих свойств на основе других моделей КП. Для обоснования этого утверждения мы приводим пример верификации КП Yahalom [94]. Верификация этого КП в вышеупомянутом источнике занимает несколько десятков страниц, в то время как верификация КП Yahalom на базе предложенной модели занимает менее 4 страниц. Причина такого существенного упрощения верификации КП связана с использованием доказанных в настоящей работе теорем 1 и 2, которые представляют собой схемы обоснования свойства защищённости сообщений при выполнении переходов протокола, и устраняют дублирование рассуждений при доказательстве корректности КП.

Предложенный формализм предназначен только для доказательства корректности криптографических протоколов, и не предназначен для обнаружения ошибок в некорректных КП. Если протокол является некорректным, то для обнаружения ошибок в нем должен использоваться другой метод, аналогичный методу Model Checking в теории верификации программ, см. [125]. Данный метод будет изложен в последующих публикациях автора.

Изложенный в статье язык описания РП имеет самостоятельную ценность, и может рассматриваться как новый язык описания распределенных алгоритмов с применением криптографических функций.

В целях простоты изложения в работе рассматривается такая математическая модель КП, которая отражает простейшие криптографические примитивы, используемые в КП: в ней формализуются лишь симметричные системы шифрования, и не рассматриваются такие примитивы как системы шифрования с открытым ключом, хэш-функции, цифровые подписи, и т.п. Все эти примитивы несложно ввести в представленную модель путем соответствующих дополнений.

2. Вспомогательные понятия

В этом параграфе мы излагаем понятия, необходимые для определения понятий последовательного и распределённого процесса.

2.1. Типы, переменные, константы и функциональные символы

Будем предполагать, что заданы следующие множества.

- Множество $Types$, его элементы называются **типами данных** (или просто **типами**). Каждому типу τ из $Types$ сопоставлено множество D_τ **значений** типа τ.
- Множества Var и Con, их элементы называются **переменными** и **константами** соответственно. Каждой переменной $x \in Var$ и константе $c \in Con$ сопоставлен тип $\tau(x)$ и $\tau(c) \in Types$ соответственно. Каждая переменная x может принимать **значения** в некотором множестве, которое будем обозначать $D_{\tau(x)}$, т.е. в различные моменты времени переменная x может быть связана с различными элементами множества $D_{\tau(x)}$.
- Множество Fun, его элементы называются **функциональными символами (ФС)**. Каждому $f \in Fun$ сопоставлен **функциональный тип (ФТ)** $\tau(f)$, который представляет собой запись вида

$$(\tau_1, \ldots, \tau_n) \to \tau, \text{ где } \tau_1, \ldots, \tau_n, \tau \in Types. \tag{1}$$

Будем считать, что среди типов, входящих в указанное выше множество $Types$, присутствуют следующие типы:

- **A**, значения этого типа называются **агентами**,
- **K**, значения этого типа обозначают **ключи**, используемые агентами для шифрования или расшифрования сообщений,
- **M**, значения этого типа обозначают **сообщения**, которые агенты могут пересылать друг другу во время своей работы,
- для каждого типа τ множество $Types$ содержит тип $\mathbf{2}^\tau$, значениями которого являются подмножества множества D_τ, тип $\mathbf{2}^\tau$ используется, например, для представления содержимого канала: его значениями могут быть множества различных сообщений, содержащихся в канале,
- для каждого списка типов $\tau_1, \ldots, \tau_n$ множество $Types$ содержит тип $(\tau_1, \ldots, \tau_n)$, и $D_{(\tau_1,\ldots,\tau_n)} = D_{\tau_1} \times \ldots \times D_{\tau_n}$.

2.2. Термы

Термы строятся из переменных, констант и ФС. Множество всех термов обозначается символом Tm. Каждый терм e имеет тип $\tau(e) \in Types$, определяемый структурой терма e.

Правила построения термов имеют следующий вид:

- если $x \in Var \cup Con$, то x – терм типа $\tau(x)$, и
- если $e_1, \ldots, e_n \in Tm$, $f \in Fun$, и $\tau(f)$ имеет вид (1), где $\tau_i = \tau(e_i)$ $(i = 1, \ldots, n)$, то запись $f(e_1, \ldots, e_n)$ – терм типа τ.

Терм $e \in Tm$ называется **подтермом** терма $e' \in Tm$, если либо $e = e'$, либо $e' = f(e_1, \ldots, e_n)$, и $\exists\, i \in \{1, \ldots, n\}$: e – подтерм терма e_i. Запись $e \subseteq e'$, где $e, e' \in Tm$, означает, что e – подтерм терма e'. Запись $e \subset e'$, где $e, e' \in Tm$, означает, что $e \subseteq e'$ и $e \neq e'$.

Индукцией по структуре терма e нетрудно доказать, что

$$\begin{array}{l}\text{если } e_1 \text{ и } e_2 \text{ – различные подтермы терма } e,\\ \text{то либо } e_1 \subset e_2, \text{ либо } e_2 \subset e_1, \text{ либо } e_1 \text{ и } e_2 \text{ не имеют общих компонентов.}\end{array} \qquad (2)$$

Будем считать, что Fun содержит следующие ФС:

- ФС *encrypt* типа $(\mathbf{K}, \mathbf{M}) \to \mathbf{M}$,
  терм вида $encrypt(k, e)$ обозначает сообщение, получаемое шифрованием сообщения e на ключе k в симметричной системе шифрования, будем обозначать такой терм записью $k(e)$ и называть его **шифрованным сообщением (ШС)**,
- ФС *shared_key* типа $(\mathbf{2^A}) \to \mathbf{K}$,
  терм вида $shared\_key(\{A_1, \ldots, A_n\})$ называется **разделяемым ключом** агентов $A_1, \ldots, A_n$ и будет обозначаться $k_{A_1,\ldots,A_n}$,
- ФС *list* типа $(\tau_1, \ldots, \tau_n) \to (\tau_1, \ldots, \tau_n)$, где $(\tau_1, \ldots, \tau_n)$ – произвольные типы, терм вида $list(e_1, \ldots, e_n)$ обозначает список термов, компоненты которого – термы $e_1, \ldots, e_n$, мы будем обозначать терм $list(e_1, \ldots, e_n)$ более короткой записью $(e_1, \ldots, e_n)$.

Отметим, что мы не предполагаем наличие в Fun ФС *decrypt*, обозначающего операцию расшифрования в симметричной системе шифрования. Это связано с тем, что операцию расшифрования шифртекста e на ключе k можно выразить в нашем языке путем действия вида $k(x) := e$, которое заключается в нахождении такого значения переменной x, результат шифрования которого на ключе k будет совпадать с значением терма e (понятие действия изложено в пункте 3).

Поясним также, каким образом в рассматриваемой модели осуществляется проверка возможности доступа участника к зашифрованной информации. Согласно определению

понятия действия вида $e := e'$, операция расшифрования может быть выполнена участником в том и только том случае когда ключ, который он использует для расшифрования, совпадает с тем ключом, на котором было зашифровано расшифровываемое сообщение. Если эти ключи разные, то расшифрование не может быть выполнено, т.е. выполнение протокола не дойдёт до своего заключительного состояния.

В предложенном формализме функциональные символы являются абстракцией идеальных криптографических примитивов, поэтому для полной гарантии корректности криптографических протоколов необходимо гарантировать корректность реализаций таких криптографических примитивов. Для гарантии корректности реализаций таких криптографических примитивов применяются методы дедуктивной верификации программ, например, дедуктивная верификация применялась для проверки корректности реализаций хэш-функций SHA-256 и «Стрибог», что описано в статьях [11] и [1] соответственно.

Также отметим, что мы не рассматриваем формализацию в нашей модели таких компонентов криптографических протоколов как асимметричное шифрование, цифровая подпись, хэш-функции, схемы разделения секрета, и т. п. Формализация всех этих компонентов является несложной и производится на основе введения дополнительных ФС.

Будем использовать следующие обозначения:

- $\forall\, x \in Var, \forall\, e \in Tm$ запись $x \in e$ означает, что x входит в e,
- $\forall\, E \subseteq Tm\ \ Var(E) = \{x \in Var \mid \exists\, e \in E : x \in e\}$,
- $\forall\, E \subseteq Tm$ запись $Tm(E)$ обозначает наименьшее по включению множество, удовлетворяющее следующим условиям:
  - $E \subseteq Tm(E)$, $Con \subseteq Tm(E)$,
  - для каждого терма вида $f(e_1, \ldots, e_n)$

    если $e_1, \ldots, e_n \in Tm(E)$, то $f(e_1, \ldots, e_n) \in Tm(E)$,
- $\forall\, \tau \in Types$, $\forall\, E \subseteq Tm\ E_\tau = \{e \in E \mid \tau(e) = \tau\}$.

2.3. Формулы и теории

Элементарной формулой (ЭФ) называется запись одного из следующих видов:

$$e = e', \quad \text{где } \tau(e) = \tau(e'),$$
$$e \in e', \quad \text{где } \tau(e') = \mathbf{2}^{\tau(e)}.$$

Формулой называется обычная булева комбинация ЭФ, в которой могут использоваться логические связки $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ и константы 1 и 0. Множество всех формул обознача-

ется Fm. Формулы вида $\wedge(e_1, e_2)$ будем записывать в более привычном виде $e_1 \wedge e_2$, аналогичная запись используется для других булевых комбинаций. Формулы вида $e_1 \wedge \ldots \wedge e_n$ могут также записываться в виде $\left\{ \begin{array}{c} e_1 \\ \ldots \\ e_n \end{array} \right\}$ или $\{e_1, \ldots, e_n\}$. Возможна конъюнкция произвольного семейства формул $\{e_i \mid i \in I\}$, она обозначается записью $\bigwedge_{i \in I} e_i$.

Теорией называется произвольная совокупность формул $Th \subseteq Fm$. Понятие **доказуемости** формулы в теории определяется стандартным образом. Ниже мы будем предполагать, что задана некоторая теория Th, и будем говорить что формула φ **доказуема**, если она доказуема в этой теории Th.

2.4. Подстановки

Подстановкой называется функция $\theta : Var \to Tm$, такая, что

$$\forall\, x \in Var \;\; \tau(x) = \tau(\theta(x)).$$

Будем говорить, что θ заменяет переменную $x \in Var$ на терм $\theta(x)$. Подстановка называется **тождественной**, если $\forall\, x \in Var \;\; \theta(x) = x$.

Будем использовать следующие обозначения:

- Θ обозначает множество всех подстановок,
- $\forall\, \theta \in \Theta,\ \forall\, e \in Tm$ запись e^θ обозначает терм, получаемый из e заменой для каждой переменной $x \in Var(e)$ каждого вхождения x в e на терм $\theta(x)$;
- $\forall\, \theta \in \Theta,\ \forall\, E \subseteq Tm\ E^\theta = \{e^\theta \mid e \in E\}$.
- $\forall\, \theta \in \Theta, \forall\, \varphi \in Fm$ запись $\theta \vdash \varphi$ обозначает утверждение, что формула φ^θ доказуема.

3. Последовательные и распределенные процессы

В этой главе определяются понятия последовательного и распределенного процесса. Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП. Предложенная модель является теоретической основой для метода верификации КП, излагаемого в пункте 6.

3.1. Действия

Действие – это запись одного из следующих видов:

$$\circ!e, \quad \circ?e, \quad e := e', \quad \text{где } e, e' \in Tm,$$

которые называются **выводом** сообщения e в открытый канал $\circ$, **вводом** сообщения e из открытого канала $\circ$, и **присваиванием**, соответственно. Множество всех действий обозначается Act.

3.2. Последовательные процессы

Последовательным процессом (или просто **процессом**) будем называть граф P со следующими свойствами:

- P имеет выделенные вершины $\odot$ и $\otimes$, называемые **начальной** и **терминальной** вершинами соответственно, из $\otimes$ не выходят рёбра,
- каждому ребру графа P сопоставлена метка $a \in Act$, ребро процесса P представляется записью $v \xrightarrow{a} v'$, где v и v' – начало и конец ребра, a – метка ребра.

Процесс является описанием поведения дискретной динамической системы, работа которой заключается в последовательном выполнении действий, связанных с вводом и выводом сообщений и изменением значений переменных. С каждым процессом P связаны

- **агент** $Agent_P \in Var_{\mathbf{A}}$, называемый **исполнителем** процесса P,
  напомним, что согласно обозначениям, введенным в конце пункта 2.2, $Var_{\mathbf{A}}$ – это множество переменных, имеющих тип **A** (агент),
- множество Var_P **переменных** процесса P, являющееся дизъюнктным объединением следующих множеств:
  - множество $Public_P$ **открытых переменных**,
  - множество $Private_P$ **приватных переменных**,
  - множество $Unique_P$ переменных, инициализированных уникальными значениями, эти переменные обозначают криптографические ключи, или переменные, называемые **нонсами**,
  - $\{x_P\}$, значения x_P – подмножества множества

$$Public_P \cup Private_P \cup Unique_P,$$

    их элементы называются **инициализированными переменными** процесса P,
  - $\{at_P\}$, значения at_P – вершины графа P,
  - $\{x_\circ\}$, $\tau(x_\circ) = \mathbf{2^M}$, значения $x_\circ$ интерпретируются как **содержимое открытого канала** (отметим, что переменная $x_\circ$ является общей для всех процессов).

В каждый момент выполнения процесса каждая его неслужебная переменная (т.е. отличная от at_P, x_P, $x_\circ$) либо инициализирована каким-либо значением, либо неинициализи-

рована. Если она в какой-либо момент времени инициализирована, то во все последующие моменты данная переменная имеет то же значение что и в момент инициализации. Переменные из $Unique_P$ инициализированы в начальный момент.

3.3. Процесс противника

Процесс противника – это процесс †, обладающий свойствами:

- множество вершин графа процесса † одноэлементно,
- $\forall\, a \in Act$ граф процесса † содержит ребро с меткой a.

Ниже будем предполагать, что † – единственный из всех рассматриваемых процессов, граф которого имеет циклы.

3.4. Распределенные процессы

Распределенным процессом (РП) называется семейство процессов $\mathcal{P} = \{P_i \mid i \in I\}$, таких, что компоненты семейства

$$\{Private_{P_i} \cup Unique_{P_i} \cup \{x_{P_i}, at_{P_i}\} \mid i \in I\} \tag{3}$$

дизъюнктны (если это условие не выполняется, то соответствующие переменные в процессах P_i переименовываются).

С каждым РП $\mathcal{P}$ связано **начальное состояние** $\theta^0_{\mathcal{P}} \in \Theta$, обладающее следующими свойствами:

$$\forall\, P \in \mathcal{P} \quad x_P^{\theta^0_{\mathcal{P}}} = Public_P \cup Unique_P,\ at_P^{\theta^0_{\mathcal{P}}} = \odot, x_\circ^{\theta^0_{\mathcal{P}}} = \emptyset.$$

Если РП состоит из одного процесса P, то он обозначается тем же символом P. Если $\{\mathcal{P}_i \mid i \in I\}$ – семейство РП, то данная запись обозначает также РП, состоящий из всех процессов, входящих в какой-либо РП из семейства $\mathcal{P}_i$ ($\forall\, i \in I$).

Записи $Public_{\mathcal{P}}$ и $Unique_{\mathcal{P}}$ обозначают соответственно множества

$$\bigcup_{P\in\mathcal{P}} Public_P \quad \text{и} \quad \bigcup_{P\in\mathcal{P}} Unique_P.$$

3.5. Переходы в распределенных процессах

Переход в РП $\mathcal{P}$ – это утверждение, обозначаемое записью $\theta \xrightarrow{a_P} \theta'$, где $P \in \mathcal{P}$, $\theta, \theta' \in \Theta$ (θ называется **началом** данного перехода, а θ' – его **концом**) и a – метка некоторого ребра $v \xrightarrow{a} v'$ процесса P, причём выполнены условия:

1. $at_P^\theta = v$, $at_P^{\theta'} = v'$,
2. $\forall\, x \in x_P^\theta \setminus \{at_P, x_P, x_\circ\}\ \ x^\theta = x^{\theta'}$,
3. если $a = \circ!e$, то
   - $e \in Tm(x_P^\theta), x_P^{\theta'} = x_P^\theta, x_\circ^{\theta'} = x_\circ^\theta \cup \{e^\theta\}$,
   - если e^θ содержит подтерм вида $k(\tilde{e})$, где $k \in Tm_{\mathbf{K}}$, то верно следующее свойство:

$$\text{либо } k \in Var, \text{ либо } \left\{ \begin{array}{l} k = shared\_key(\ldots) \\ Agent_P \in k \end{array} \right\}, \tag{4}$$

   напомним, что согласно обозначениям, введенным в конце пункта 2.2, $Tm_{\mathbf{K}}$ – это множество термов, имеющих тип **K** (ключ), и $Agent_P \in k$ означает, что переменная $Agent_P$ входит в терм k,
4. если $a = \circ?e$ или $e := e'$, то
   (a) $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(e)$,
   (b) $\theta' \vdash e \in x_\circ$ или $\left\{ \begin{array}{l} \theta' \vdash e = e' \\ e' \in Tm(x_P^\theta) \end{array} \right\}$ соответственно,
   (c) если e^θ содержит подтерм вида $k(\tilde{e})$, где $k \in Tm_{\mathbf{K}}$, то верно свойство (4),
   (d) если $a = (e := e')$ и $k = shared\_key(\ldots)$, то верна импликация

$$k \subseteq (e')^\theta \ \Rightarrow\ Agent_P \in k. \tag{5}$$

Переход $\theta \xrightarrow{a_P} \theta'$ РП $\mathcal{P}$ интерпретируется как выполнение процессом $P \in \mathcal{P}$ действия a, в результате чего $\mathcal{P}$ переходит от θ к θ'. Если в текущий момент с $\mathcal{P}$ связана подстановка θ, и в этот момент некоторый процесс P, входящий в $\mathcal{P}$, содержит ребро $v \xrightarrow{a} v'$, причем $v = at_P^\theta$, то мы считаем, что РП $\mathcal{P}$, связанный в текущий момент с подстановкой θ, может выполнить действие a, после чего он будет связан с подстановкой θ', удовлетворяющей вышеприведённым условиям при этом

- при выполнении действия $\circ!e$ происходит добавление терма e^θ к содержимому открытого канала $\circ$,
- при выполнении действия $\circ?e$ или $e := e'$ происходит либо чтение некоторого терма из содержимого канала $\circ$, либо присваивание соответственно, путем инициализации неинициализированных в текущий момент переменных из терма e: терм e рассматривается как шаблон, которому должен соответствовать некоторый терм из $x_\circ^\theta$ или терм $(e')^\theta$ соответственно, и выполняемое действие заключается в преобразовании θ в подстановку θ' путем определения подходящих значений переменных из $Var(e) \setminus x_P^\theta$,

с таким расчётом, чтобы значение $e^{\theta'}$ было бы равно некоторому терму из $x_\circ^\theta$ или терму $(e')^\theta$ соответственно.

3.6. Выполнение распределенного процесса

Выполнение РП $\mathcal{P}$ – это последовательность подстановок $\pi = (\theta_0, \theta_1, \ldots)$ РП $\mathcal{P}$, такая, что θ_0 – начальное состояние РП $\mathcal{P}$, и для каждой пары θ_i, θ_{i+1} соседних членов этой последовательности имеется переход $\theta_i \overset{a_P}{\to} \theta_{i+1}$, где P – какой-либо процесс из $\mathcal{P}$.

Для каждого выполнения $\pi = (\theta_0, \theta_1, \ldots)$ запись $\theta \in \pi$ означает, что $\exists\, i \geq 0 : \theta_i = \theta$.

Если задано выполнение $\pi = (\theta_0, \theta_1, \ldots)$ и θ, θ' – подстановки, входящие в π, то запись $\theta <_\pi \theta'$ означает, что $\theta = \theta_i$ и $\theta' = \theta_j$ для некоторых индексов $i < j$. Запись $\theta \leq_\pi \theta'$ означает, что $\theta <_\pi \theta'$ или $\theta = \theta'$.

Подстановка θ РП $\mathcal{P}$ называется **достижимым состоянием** РП $\mathcal{P}$, если она входит в некоторое выполнение $\mathcal{P}$. Множество всех достижимых состояний РП $\mathcal{P}$ обозначается $\Theta_\mathcal{P}$.

В начальный момент выполнения РП $\mathcal{P}$ переменные из $Unique_\mathcal{P}$ инициализированы **уникальными значениями**, т.е. такими значениями, которые никогда не встречались среди всех значений, используемых до начала выполнения $\mathcal{P}$.

4. Свойство защищённости

4.1. Определение свойства защищённости

В рассуждениях, связанных с верификацией РП, будем использовать свойство **защищённости**, обозначаемое записью

$$E \perp P, \text{ где } \begin{cases} E \subseteq Public_\mathcal{P} \cup Tm(Public_\mathcal{P})_\mathbf{K}, \\ P \in \mathcal{P}, \forall\, k \in E_\mathbf{K} \;\; Agent_P \notin k, \end{cases} \tag{6}$$

где $\mathcal{P}$ – некоторый РП. Напомним, что согласно обозначениям, введенным в конце пункта 2.2, $E_\mathbf{K}$ – это множество термов из E типа $\mathbf{K}$ (ключ).

Свойство (6) истинно в состоянии $\theta \in \Theta_\mathcal{P}$ (что обозначается записью $\theta \models E \perp P$), если

$$\begin{gathered} \forall\, e \in E,\ \forall\, e' \in (x_P^\theta)^\theta \cup x_\circ^\theta \text{ каждое вхождение } e \text{ в } e' \\ \text{содержится в подтерме } k(\ldots) \subseteq e',\ \text{где } k \in E_\mathbf{K}. \end{gathered} \tag{7}$$

Данное свойство имеет следующий смысл: термы из E доступны процессу P в состоянии θ только в «защищённом» виде, т.е. содержатся в термах из $(x_P^\theta)^\theta \cup x_\circ^\theta$ только внутри ШС, которые зашифрованы на ключах, недоступных для P в θ.

4.2. Сохранение защищённости при переходах

В этом параграфе доказывается теорема о сохранении свойства защищённости $E \perp P$ при переходах РП. Данная теорема м.б. интерпретирована как следующее утверждение: если в текущем состоянии θ верно свойство $E \perp P$, то никакая собственная активность процесса P, начиная с состояния θ, не приведет к тому, что какое-либо сообщение из E когда-нибудь станет доступным процессу P.

Теорема 1

Пусть задан переход $\theta \xrightarrow{a_P} \theta'$ в РП $\mathcal{P}$.

$\forall\, E \subseteq Public_{\mathcal{P}} \cup Tm(Public_{\mathcal{P}})_{\mathbf{K}}$ верна импликация

$$\theta \models E \perp P \Rightarrow \theta' \models E \perp P. \tag{8}$$

Доказательство.

Пусть верна посылка импликации (8), т.е. верно (7). Докажем, что тогда будет верно её заключение, т.е.

$$\begin{array}{l} \forall\, e \in E,\ \forall\, e' \in (x_P^{\theta'})^{\theta'} \cup x_\circ^{\theta'} \text{ каждое вхождение } e \text{ в } e' \\ \text{содержится в подтерме } k(\tilde{u}) \subseteq e',\ \text{где } k \in E_{\mathbf{K}}. \end{array} \tag{9}$$

Если (9) неверно, то

$$\begin{array}{l} \exists\, e \in E, \exists\, e' \in (x_P^{\theta'})^{\theta'} \cup x_\circ^{\theta'}, \exists \text{ вхождение } e \subseteq e',\ \text{которое} \\ \text{не содержится в каждом терме вида } k(\tilde{u}) \subseteq e', \text{где } k \in E_{\mathbf{K}}. \end{array} \tag{10}$$

Рассмотрим три варианта перехода $\theta \xrightarrow{a_P} \theta'$.

1. $a_P = \circ!u$, $u \in Tm(x_P^\theta), x_P^{\theta'} = x_P^\theta, x_\circ^{\theta'} = x_\circ^\theta \cup \{u^\theta\}$.

   В этом случае (10) возможно только если $e' = u^\theta$.

   Возможен один из следующих двух случаев:

   (a) $u \in x_P^\theta$, в этом случае из $e \subseteq u^\theta = e'$ и (7) следует

   $$e \subseteq k(\tilde{u}) \subseteq u^\theta = e', \quad \text{где } k \in E_{\mathbf{K}},$$

   что противоречит (10),

   (b) $u = f(u_1, \ldots, u_n)$, где $f \in Fun$, в этом случае противоречивость (10) обосновывается индукцией по структуре u: в случае $e \subset u^\theta$ противоречивость (10) следует

из индуктивного предположения, а в случае $e = u^\theta = e'$, согласно свойству (7), e содержится в подтерме $k(\tilde{u}) \subseteq e' = e$, т.е. $e = k(\tilde{u})$, что невозможно по условию из (6) на множество E.

2. $a_P = \circ?u$, $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(u)$, $u^{\theta'} \in x_\circ^\theta$, и если u^θ содержит подтерм $k(\tilde{u})$, где $k \in Tm_{\mathbf{K}}$, то верно (4).

   В этом случае (10) возможно только если

$$\begin{aligned} &\exists\, e \in E,\ \exists\, y \in Var(u) \setminus x_P^\theta : e \subseteq y^{\theta'} = e', \\ &\text{и не существует терма вида } k(\tilde{u}), \text{ где } k \in E_{\mathbf{K}}, \\ &\text{такого, что } e \subseteq k(\tilde{u}) \subseteq y^{\theta'}. \end{aligned} \tag{11}$$

   Поскольку упомянутый в (11) терм e содержится в терме $y^{\theta'} \subseteq u^{\theta'} \in x_\circ^\theta$, то из (7) следует, что e содержится в некотором подтерме вида $k(\tilde{u}) \subseteq u^{\theta'}$, где $k \in E_{\mathbf{K}}$.

   Таким образом, $e \subseteq k(\tilde{u})$ и $e \subseteq y^{\theta'}$, т.е. термы $k(\tilde{u})$ и $y^{\theta'}$ имеют непустое пересечение, поэтому, согласно (2), либо $k(\tilde{u}) \subseteq y^{\theta'}$, либо $y^{\theta'} \subset k(\tilde{u})$. Включение $k(\tilde{u}) \subseteq y^{\theta'}$ противоречит (11), следовательно $y^{\theta'} \subset k(\tilde{u})$, поэтому

$$y^{\theta'} \subset k(\tilde{u}) \subseteq u^{\theta'}. \tag{12}$$

   Докажем индукцией по структуре терма u, что из правого включения в (12) следует, что

$$\exists\, z \in Var(u) : k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{13}$$

   Если $u \in Var$, то $z = u$, если $u \in Con$, то (12) неверно.

   Пусть $u = f(u_1, \ldots, u_n)$, где $f \in Fun$, тогда возможны следующие случаи:

   - $f = encrypt$, т.е. $u = k_1(u_1)$: если $k_1 \in Var_{\mathbf{K}}$, то возможны следующие случаи:
     - $k(\tilde{u}) = u^{\theta'} = k_1^\theta(u_1^{\theta'})$, в этом случае $k = k_1^\theta \in E$, и, согласно (4), либо $k \in Var$, либо

$$k = shared\_key(\ldots) \text{ и } Agent_P \in k. \tag{14}$$

       Случай $k \in Var$ невозможен, т.к. из свойств $k \in (x_P^\theta)^\theta$ и $k \in E_{\mathbf{K}}$, согласно (7), следует, что вхождение k в k должно содержаться в подтерме вида $k'(\ldots) \subseteq k$, что невозможно, а (14) невозможно согласно второй строке в (6),
     - $k(\tilde{u}) \subseteq k_1^{\theta'}$, данный случай невозможен по определению термов типа $\mathbf{K}$,
     - $k(\tilde{u}) \subseteq u_1^{\theta'}$, в данном случае утверждение (13) следует из индуктивного предположения,

а если $k_1 = shared\_key(\ldots)$, то верно (14), что невозможно согласно второй строке в (6),

- если $f = list$, то $\exists\, i \in 1, \ldots, n : k(\tilde{u}) \subseteq u_i^{\theta'}$, и (13) следует из индуктивного предположения,
- случай $f = shared\_key$ невозможен.

Из (12) и (13) следует, что

$$y^{\theta'} \subset k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{15}$$

Таким образом, u содержит вхождения переменных y и z, обладающие следующим свойством: $y^{\theta'} \subset z^{\theta'}$, откуда для данных вхождений следует включение $y \subset z$, что невозможно.

3. $a_P = (u := u')$, $x_\circ^{\theta'} = x_\circ^\theta$, $x_P^{\theta'} = x_P^\theta \cup Var(u)$, $u' \in Tm(x_P^\theta)$, $u^{\theta'} = (u')^\theta$, и если u^θ содержит подтерм вида $k(\tilde{u})$, где $k \in Tm_{\mathbf{K}}$, то верно (4).

Анализ данного случая аналогичен анализу предыдущего случая. Свойство (10) верно только если верно свойство (11), из которого следует, что терм e, упомянутый в (11), обладает свойством $e \subseteq u^{\theta'} = (u')^\theta$.

Докажем, что

$$\exists\, v \in Var(u') : e \subseteq v^\theta. \tag{16}$$

Т.к. $e \in E$, то, согласно (6),

- либо $e \in Var$, в этом случае (16) очевидно,
- либо e имеет вид $shared\_key(\ldots)$, в этом случае из $e \subseteq (u')^{\theta'}$, соотношения (5) и второй строки в (6) следует, что данный случай невозможен.

Т.к. $u' \in Tm(x_P^\theta)$, т.е. $Var(u') \subseteq x_P^\theta$, то из (16) следует, что $v \in x_P^\theta$, и $e \subseteq v^\theta \in (x_P^\theta)^\theta$, откуда, на основании (7), заключаем, что $e \subseteq k(\tilde{u}) \subseteq v^\theta \subseteq (u')^\theta = u^{\theta'}$, где $k \in E_{\mathbf{K}}$.

Термы $k(\tilde{u})$ и $y^{\theta'}$ имеют непустое пересечение (оба содержат e), и из (11) следует, что $k(\tilde{u})$ не м.б. подтермом терма $y^{\theta'}$, поэтому из (2) следует, что

$$y^{\theta'} \subset k(\tilde{u}) \subseteq u^{\theta'}. \tag{17}$$

Так же, как и в предыдущем случае, доказываем, что из (17) следует свойство

$$\exists\, z \in Var(u) : k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{18}$$

Из (17) и (18) следует, что

$$y^{\theta'} \subset k(\tilde{u}) \subseteq z^{\theta'} \subseteq u^{\theta'}. \tag{19}$$

Таким образом, u содержит вхождения переменных y и z, обладающие следующим свойством: $y^{\theta'} \subset z^{\theta'}$, откуда для данных вхождений следует включение $y \subset z$, что невозможно. ■

5. Свойство соответствия

В этом параграфе формулируется и доказывается теорема, которая может использоваться для обоснования **свойства соответствия** протоколов аутентификации. Данное свойство имеет следующий неформальный смысл: если один из участников протокола аутентификации (обозначим его A) после выполнения этого протокола пришел к выводу, что другой участник этого протокола (обозначим его B) является подлинным (т.е. те параметры, которые получил A от якобы участника B, совпадают с теми параметрами, которые B посылал A), то B действительно посылал A сообщение с этими параметрами.

Доказываемая ниже теорема имеет следующий смысл: если при некотором выполнении π РП $\mathcal{P}$ в состоянии $\theta \in \pi$ в канале $\circ$ содержится сообщение, содержащее подтерм $k(e)$, где ключ k недоступен в состоянии θ для некоторого процесса $P \in \mathcal{P}$, то в некотором состоянии $\theta' <_\pi \theta$ другой процесс $P' \neq P$ из $\mathcal{P}$ послал в канал $\circ$ сообщение, содержащее $k(e)$.

Теорема 2

Пусть заданы

- РП $\mathcal{P}$ и некоторое его выполнение $\pi = (\theta_0, \theta_1, \ldots)$,
- подмножество $E \subseteq Public_{\mathcal{P}} \cup Tm(Public_{\mathcal{P}})_{\mathbf{K}}$, и
- состояние $\theta \in \pi$, причем $\theta \models E \bot P$, и $\exists\, e \in x_\circ^\theta$:

$$\exists\, k(\tilde{e}) \subseteq e, \text{ где } k \in E_{\mathbf{K}}. \tag{20}$$

Тогда $\exists\, P' \in \mathcal{P} : P' \neq P$ и π содержит переход вида

$$\dot{\theta} \overset{(\circ!\dot{e})_{P'}}{\longrightarrow} \theta', \text{ где } k(\tilde{e}) \subseteq \dot{e}^{\dot{\theta}} \text{ и } \theta' \leq_\pi \theta. \tag{21}$$

Доказательство.

Пусть θ' – первое состояние на π, такое, что $\exists\, e' \in x_\circ^{\theta'}$:

$$k(\tilde{e}) \subseteq e', \text{ где } k(\tilde{e}) \text{ – терм из (20)}. \tag{22}$$

Нетрудно видеть, что $\theta' \leq_\pi \theta$, и т.к. $x_\circ^{\theta_0} = \emptyset$, то $\theta' \neq \theta_0$.

Пусть переход на пути π с концом в θ' имеет вид $\dot{\theta} \xrightarrow{a_{P'}} \theta'$. Т.к. $e' \notin x^{\dot{\theta}}_{\circ}$, то $a'_P = \circ!\dot{e}$, где $\dot{e}^{\dot{\theta}} = e'$. Если $P' \neq P$, то теорема доказана.

Докажем, что случай $P' = P$ невозможен.

Пусть $P' = P$, т.е. $\dot{\theta} \xrightarrow{(\circ!\dot{e})_P} \theta'$.

Если $k = shared\_key(\ldots)$, то, согласно (22) и (4), $Agent_P \in k \in E_{\mathbf{K}}$, что противоречит второй строке в (6). Поэтому $k \in Var$.

Из $\theta \models E \bot P$ следует, что $\dot{\theta} \models E \bot P$.

Случай $k \in (x^{\dot{\theta}}_P)^{\dot{\theta}}$ невозможен, т.к. из $k \in (x^{\dot{\theta}}_P)^{\dot{\theta}}$ и $k \in E_{\mathbf{K}}$, согласно (7), следует, что вхождение k в k должно содержаться в подтерме вида $k'(\ldots) \subseteq k$, что невозможно.

Докажем индукцией по структуре терма $\dot{e}$, что из свойства

$$k(\tilde{e}) \subseteq \dot{e}^{\dot{\theta}}, k \in E_{\mathbf{K}}, k \in Var \text{ и } k \notin (x^{\dot{\theta}}_P)^{\dot{\theta}} \tag{23}$$

следует утверждение

$$\exists\, x \in Var(\dot{e}) \subseteq x^{\dot{\theta}}_P : k(\tilde{e}) \subseteq x^{\dot{\theta}}. \tag{24}$$

Если $\dot{e} \in Var$, то $x = \dot{e}$, а если $\dot{e} \in Con$, то (23) неверно.

Пусть $\dot{e} = f(e_1, \ldots, e_n)$, где $f \in Fun$, тогда

- если $f = encrypt$, т.е. $\dot{e} = k_1(e_1)$, то возможны следующие случаи:
  - $k(\tilde{e}) = \dot{e}^{\dot{\theta}} = k^{\dot{\theta}}_1(e^{\dot{\theta}}_1)$, в этом случае
    - $k = k^{\dot{\theta}}_1$, и если k_1 имеет вид $shared\_key(\ldots)$, то $k = k^{\dot{\theta}}_1$ тоже имеет такой вид, но по предположению (23) $k \in Var$, и
    - если $k_1 \in Var$, то $k_1 \in x^{\dot{\theta}}_P$, т.е. $k = k^{\dot{\theta}}_1 \in (x^{\dot{\theta}}_P)^{\dot{\theta}}$, что противоречит (23),
  - $k(\tilde{e}) \subseteq k^{\dot{\theta}}_1$, данный случай невозможен по определению термов типа $\mathbf{K}$,
  - $k(\tilde{e}) \subseteq u^{\dot{\theta}}_1$, в данном случае утверждение (13) следует из индуктивного предположения,
- если $f = list$, то $\exists\, i \in 1, \ldots, n$: $k(\tilde{e}) \subseteq e^{\dot{\theta}}_i$, и (13) следует из индуктивного предположения,
- случай $f = shared\_key$ невозможен.

Таким образом, из (24) следует, что $\exists\, x \in x^{\dot{\theta}}_P : k(\tilde{e}) \subseteq x^{\dot{\theta}}$.

Пусть θ'' – первое состояние на пути π, такое, что $(x^{\theta''}_P)^{\theta''}$ содержит терм с подтермом $k(\tilde{e})$, т.е.

$$\exists\, x \in x^{\theta''}_P : k(\tilde{e}) \subseteq x^{\theta''}. \tag{25}$$

Из (24) следует, что $\theta'' \leq_\pi \ddot{\theta}$. Нетрудно видеть, что θ'' – не начальное состояние, поэтому на пути π существует ребро вида $\ddot{\theta} \xrightarrow{a_{P''}} \theta''$. Обозначим $a = a_{P''}$. Из определения θ'' следует, что $x \notin x_P^{\ddot{\theta}}$, поэтому $P'' = P$ (т.к. при этом переходе изменяется значение x_P), и возможны два случая:

1. $a = \circ?\ddot{e}$, $x \in Var(\ddot{e})$, $\ddot{e}^{\theta''} \in x_\circ^{\ddot{\theta}}$,

   т.к. $k(\tilde{e}) \subseteq x^{\theta''} \subseteq \ddot{e}^{\theta''} \in x_\circ^{\ddot{\theta}}$, то получаем противоречие с выбором θ' как самого первого состояния на пути π, такого, что $x_\circ^{\theta'}$ содержит терм e' с подтермом $k(\tilde{e})$: $\ddot{\theta}$ имеет аналогичное свойство, и находится левее θ',

2. $a = (\ddot{e} := \bar{e})$, $x \in Var(\ddot{e})$, $\bar{e} \in Tm(x_P^{\ddot{\theta}})$, $\ddot{e}^{\theta''} = \bar{e}^{\ddot{\theta}}$,

   поскольку

   - $k(\tilde{e}) \subseteq x^{\theta''} \subseteq \ddot{e}^{\theta''} = \bar{e}^{\ddot{\theta}}$ и
   - согласно (23), $k \notin (x_P^{\dot{\theta}})^{\dot{\theta}}$, поэтому из $\ddot{\theta} <_\pi \dot{\theta}$ следует, что $k \notin (x_P^{\ddot{\theta}})^{\ddot{\theta}}$ (т.к. $x_P^{\ddot{\theta}} \subseteq x_P^{\dot{\theta}}$),

   то, аналогично доказательству импликации (23) $\Rightarrow$ (24) (заменяя в ней $\dot{e}$ на $\bar{e}$ и $\dot{\theta}$ на $\ddot{\theta}$) можно доказать утверждение

   $$\exists\, x \in Var(\bar{e}) \subseteq x_P^{\ddot{\theta}} : k(\tilde{e}) \subseteq x^{\ddot{\theta}},$$

   которое противоречит выбору θ'' как самого первого состояния на пути π со свойством (25): $\ddot{\theta}$ имеет аналогичное свойство, и находится левее θ''. ■

6. Метод верификации протоколов, основанный на представленной модели

6.1. Описание метода верификации

Изложенная в предыдущих пунктах модель криптографических протоколов может применяться для обоснования таких свойств протоколов, которые представляют собой утверждения следующего типа: если при каком-либо выполнении π анализируемого протокола он достиг некоторого состояния $\theta \in \pi$, то существуют состояния $\theta', \ldots \leq_\pi \theta$ на этом выполнении, которые обладают заданными свойствами. В этом пункте в качестве такого свойства рассматривается свойство соответствия в протоколах аутентификации, определяемое ниже. Метод обоснования этого свойства заключается в обратном построении выполнения данного протокола, начиная с состояния θ. Искомые состояния $\theta', \ldots \leq_\pi \theta$ возникают в процессе обратного построения выполнения протокола. Построение данного выполнения производится с использованием теоремы 2.

Ниже излагается иллюстрация применения данного метода для верификации свойств соответствия и секретности протокола аутентификации Yahalom.

6.2. Описание протокола Yahalom

Протокол Yahalom предназначен для аутентификации (т.е. проверки подлинности) агентов, взаимодействующих по открытому каналу $\circ$, и передачи сеансовых ключей между этими агентами. Предполагается что

- заданы множество агентов Ag, а также агент J, называемый **доверенным посредником**, данные агенты могут взаимодействовать друг с другом по открытому каналу $\circ$,
- каждый агент $A \in Ag$ имеет разделяемый ключ k_{AJ} с доверенным посредником J, на котором A и J могут шифровать и расшифровывать сообщения, используя симметричную систему шифрования.

В каждом сеансе протокола Yahalom принимают участие следующие агенты: **инициатор** $A \in Ag$, **доверенный посредник** J, и **респондер** $B \in Ag$. Каждый агент из Ag в одних сеансах м.б. инициатором, а в других – респондером. Выполнение сеанса протокола Yahalom с инициатором A, респондером B и доверенным посредником J представляет собой совокупность четырех пересылок сообщений:

$$\begin{array}{lll} 1. \quad A \to B & : & A, n_A \\ 2. \quad B \to J & : & B, k_{BJ}(A, n_A, n_B) \\ 3. \quad J \to A & : & k_{AJ}(B, k, n_A, n_B), k_{BJ}(A, k) \\ 4. \quad A \to B & : & k_{BJ}(A, k), k(n_B) \end{array} \tag{26}$$

Пересылки в (26) имеют следующий смысл.

1. A посылает B запрос на аутентификацию и генерацию сеансового ключа k, этот запрос состоит из имени агента A и нонса n_A.
2. B посылает J запрос на генерацию сеансового ключа k, в свой запрос он включает своё имя, имя агента A, для связи с которым нужен этот ключ, полученный нонс n_A, и свой нонс n_B.
3. J генерирует сеансовый ключ k и посылает A пару сообщений, из первого сообщения A может извлечь сеансовый ключ k, а второе предназначено для того, чтобы A переслал его B.
4. A посылает B пару сообщений,

- первое из которых было получено им от J, B может извлечь из этого сообщения сеансовый ключ k, и
- используя ключ k, B расшифровывает второе сообщение.

Если результат расшифрования совпадает с n_B, то это является для B доказательством того, что отправителем этого сообщения был A.

6.3. Некоторые определения и обозначения

1. Для каждого процесса P запись P^* обозначает РП $\{P_i \mid i \in I\}$, где I – множество натуральных чисел, и $\forall\, i \in I \;\; P_i = P$.

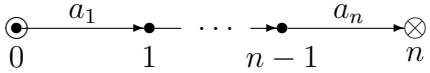
   Будем использовать следующее соглашение:

   - если в каком-либо рассуждении, связанном с РП вида P^*, некоторый процесс является первым из рассматриваемых процессов, входящих в P^*, то этот процесс и все его переменные обозначаются теми же записями, которые используются в P,
   - если кроме этого процесса рассматривается другой процесс, входящий в P^* (возможно совпадающий с P), то он обозначается P_1, и в обозначениях тех его переменных, которые являются дубликатами переменных из множества

   $$Unique_P \cup Private_P \cup \{at_P, x_P\},$$

   используется индекс 1 (например, дубликат переменной x в P_1 будет обозначаться записью x_1), в следущем процессе (P_2, который возможно совпадает с P или P_1) соответствующие переменные будут обозначаться с индексом 2, и т.д.

2. Процесс называется **линейным**, если он имеет вид



(27)

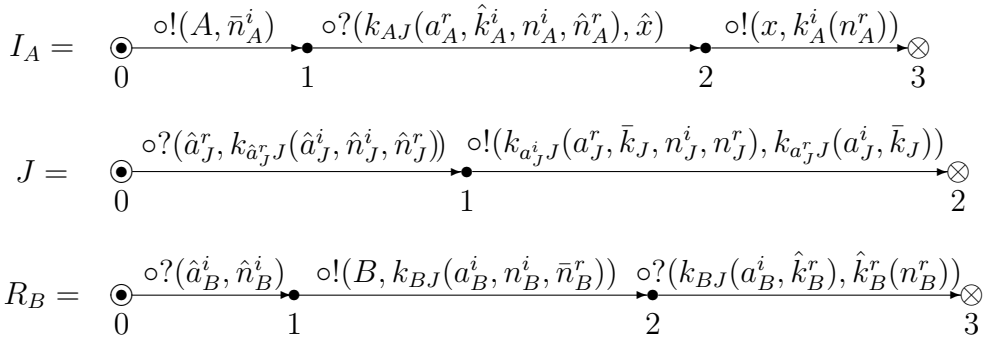
   В целях большей наглядности будем использовать следующее соглашение в обозначениях переменных в линейных процессах: пусть P – процесс вида (27), и переменная x входит в действие a_i, причём $\forall\, j \in \{1, \ldots, i-1\}$ x не входит в a_j, тогда

   - если $x \in Unique_P$, то будем указывать горизонтальную черту над всеми вхождениями x в a_i (т.е. обозначать их $\bar{x}$) и
   - если $x \in Private_P$, то будем указывать уголок над всеми вхождениями x в a_i (т.е. обозначать их $\hat{x}$).

3. Если π – выполнение РП $\mathcal{P}$, то запись $\pi \ni P^{i,i'} : \theta \xrightarrow{a} \theta'$ имеет следующий смысл: π содержит переход $\theta \xrightarrow{a_P} \theta'$, и $at_P^{\theta} = i, at_P^{\theta'} = i'$.

6.4. Формальное описание процессов, входящих в протокол Yahalom

Описание сеанса протокола Yahalom изображается схемой



В этой схеме первая и третья диаграммы соответствуют процессам I_A и R_B, описывающим поведение инициатора A и респондера B соответственно, вторая диаграмма соответствует процессу, описывающему поведение посредника J, этот процесс обозначается символом J. Верхний индекс i или r при какой-либо переменной означает, что она содержит информацию об инициаторе (i) или респондере (r) данного сеанса. Смысл переменных в этих процессах усматривается из сопоставления действий в этих процессах с соответствующими действиями в (26). Предполагаем, что $Agent_{I_A} = A$, $Agent_{R_B} = B$, $Agent_J = J$.

РП $\mathcal{P}$, соответствующий протоколу Yahalom, имеет вид

$$\mathcal{P} = \{\{I_A^* \mid A \in Ag\}, \{R_B^* \mid B \in Ag\}, J^*, \dagger\}, \tag{28}$$

т.е. каждый агент может участвовать в неограниченном числе сеансов как в качестве инициатора, так и в качестве респондера.

6.5. Свойства протокола Yahalom

В этом пункте приводится формальное описание и верификация трех свойств протокола (28): секретность ключей k_J и нонсов n_B^r, аутентификация инициатора перед респондером и аутентификация респондера перед инициатором. В доказательствах теорем 3, 4, 5 при каждом применении теоремы 2 имеется единственный вариант обоснования существования перехода (21), и мы будем сразу будем излагать это обоснование, без упоминания о

единственности варианта такого обоснования.

Теорема 3 (секретность ключей k_J и нонсов n^r_B)

РП (28) обладает следующим свойством:

$$\forall\, \theta \in \Theta_{\mathcal{P}}\ \ \theta \models E\bot\dagger,\ \text{где}\ E = \{k_{BJ}, k_J, n^r_B \mid B \in Ag\}. \tag{29}$$

Доказательство.

Докажем (29) от противного.

Предположим, что задано выполнение $\pi = (\theta_0, \theta_1, \ldots)$, и

$$S = \{\theta \in \pi \mid \theta \not\models E\bot\dagger\} \neq \emptyset.$$

Пусть θ – первое состояние на π, которое принадлежит S.

Т.к. $\theta_0 \models E\bot\dagger$, то $\theta \neq \theta_0$, т.е. в π есть переход вида $\theta' \xrightarrow{a_P} \theta$.

Из определения θ следует, что $\theta' \models E\bot\dagger$, $\theta \not\models E\bot\dagger$.

Если бы было верно $P = \dagger$, то, согласно теореме 1, отсюда следовало бы, что $\theta \models E\bot\dagger$, что противоречит определению θ.

Таким образом, $P \in \{I_A, R_B, J \mid A, B \in Ag\}$, и нетрудно видеть, что a_P имеет вид $\circ!e$, $x^{\theta}_{\circ} = x^{\theta'}_{\circ} \cup \{e^{\theta'}\}$, и верно утверждение $\theta \not\models E\bot\dagger$, которое в данной ситуации эквивалентно утверждению

$$\begin{array}{l}\exists\, u \in E,\ \exists\ \text{вхождение } u \text{ в } e^{\theta'}, \text{ не содержащееся}\\ \text{ни в каком подтерме вида } k(\ldots) \subseteq e^{\theta'}, \text{ где } k \in E_{\mathbf{K}}.\end{array} \tag{30}$$

Перебором всех возможных обоснований перехода $\theta' \xrightarrow{\circ!e} \theta$ со свойством (30) находим единственное обоснование:

$$\pi \ni I^{2,3}_A : \theta' \xrightarrow{\circ!e} \theta,\ \ \text{где}\ e = (x, k^i_A(n^r_A)), \tag{31}$$

поэтому (30) можно переписать следующим образом:

$$\begin{array}{l}\exists\, u \in E,\ \ \exists\ \text{вхождение } u \text{ в } e^{\theta'} = (x, k^i_A(n^r_A))^{\theta'},\\ \text{не содержащееся ни в каком подтерме}\\ \text{вида } k(\ldots) \subseteq (x, k^i_A(n^r_A))^{\theta'}, \text{ где } k \in E_{\mathbf{K}}.\end{array} \tag{32}$$

Т.к. $\theta' \vdash at_{I_A} = 2$, то $\exists\, \theta_1 \leq_\pi \theta'$:

$$\pi \ni I^{1,2}_A : \theta'_1 \xrightarrow{\circ?e_1} \theta_1,\ \text{где}\ e_1 = (k_{AJ}(a^r_A, \hat{k}^i_A, n^i_A, \hat{n}^r_A), \hat{x}). \tag{33}$$

Т.к. $\theta'_1 <_\pi \theta_1 \leq_\pi \theta'$ и $\theta' \models E\bot\dagger$, то

$$\theta'_1 \models E\bot\dagger. \tag{34}$$

Из (33) следует, что $e_1^{\theta'_1} = (\ldots, x^{\theta'_1}) \in x_\circ^{\theta'_1}$ (многоточия в терме $(\ldots, x^{\theta'_1})$ и ниже обозначают компоненты, не представляющие интерес для рассмотрения), поэтому, согласно (34) и (7), верно утверждение:

$$\begin{array}{l} \forall\, u \in E \text{ каждое вхождение } u \text{ в } x^{\theta'_1} \subseteq e_1^{\theta'_1} \\ \text{содержится в подтерме } k(\ldots) \subseteq x^{\theta'_1}, \text{ где } k \in E_{\mathbf{K}}. \end{array} \tag{35}$$

Сопоставляя (32) и (35), заключаем:

$$\begin{array}{l} \exists\, u \in E, \exists \text{ вхождение } u \text{ в } (k_A^i(n_A^r))^{\theta'}, \\ \text{не содержащееся ни в каком подтерме} \\ \text{вида } k(\ldots) \subseteq (k_A^i(n_A^r))^{\theta'}, \text{ где } k \in E_{\mathbf{K}}. \end{array} \tag{36}$$

По теореме 2, из (34), $e_1^{\theta'_1} \in x_\circ^{\theta'_1}$ и $k_{AJ} \in E$ следует, что $\exists\, \theta_2 \leq_\pi \theta'_1 : \pi$ содержит переход $\theta'_2 \overset{(\circ!e_2)_P}{\longrightarrow} \theta_2$, где $P \in \mathcal{P}$ и первая компонента $k_{AJ}(\ldots)$ терма $e_1^{\theta'_1}$ входит в $e_2^{\theta'_1}$. Перебором всех вариантов такого перехода находим единственное обоснование:

$$\begin{cases} \pi \ni J^{1,2} : \theta'_2 \overset{\circ!e_2}{\to} \theta_2, \text{ где } e_2 = (k_{a_J^i J}(a_J^r, \bar{k}_J, n_J^i, n_J^r), \ldots) \\ k_{(a_J^i)^\theta J}((a_J^r)^\theta, \bar{k}_J, \ldots) = k_{AJ}(a_A^r, (k_A^i)^\theta, \ldots) \end{cases} \tag{37}$$

Из второй строчки в (37) следует, что $(k_A^i)^\theta = k_J$, поэтому из (36) следует утверждение:

$$\begin{array}{l} \exists\, u \in E, \exists \text{ вхождение } u \text{ в } k_J(\ldots), \text{ не содержащееся} \\ \text{ни в каком подтерме вида } k(\ldots) \subseteq k_J(\ldots), \text{ где } k \in E_{\mathbf{K}}, \end{array}$$

которое, очевидно, противоречиво. ■

Теорема 4 (аутентификация инициатора перед респондером)

РП (28) обладает следующим свойством: $\forall\, R_B \in \mathcal{P}$, $\forall\, \theta \in \Theta_{\mathcal{P}}$, если $\theta \vdash at_{R_B} = 3$, то $\exists\, I_A \in \mathcal{P}$:

$$\theta \vdash \left\{ \begin{array}{l} at_{I_A} = 3 \\ a_A^r = B,\ a_B^i = A \\ n_A^i = n_B^i,\ n_A^r = n_B^r \\ k_A^i = k_B^r \end{array} \right\} \tag{38}$$

Доказательство.

Пусть процесс $R_B \in \mathcal{P}$ и состояние $\theta \in \Theta_{\mathcal{P}}$ таковы, что $\theta \vdash at_{R_B} = 3$, и $\pi = (\theta_0, \theta_1, \ldots)$ – выполнение РП $\mathcal{P}$, такое, что $\theta \in \pi$.

Из $\theta \vdash at_{R_B} = 3$ следует: $\exists\, \theta_1 \leq_\pi \theta$:

$$\pi \ni R_B^{2,3} : \theta_1' \overset{\circ ? e_1}{\to} \theta_1, \text{ где } e_1 = (k_{BJ}(a_B^i, \hat{k}_B^r), \hat{k}_B^r(n_B^r)).$$

По теореме (3) верно (29). Согласно теореме (2), в этом случае из $e_1^\theta \in x_\circ^{\theta_1}$ и $k_{BJ} \in E$ следует: $\exists\, \theta_2 \leq_\pi \theta_1'$:

$$\begin{cases} \pi \ni J^{1,2} : \theta_2' \overset{\circ ! e_2}{\to} \theta_2, \text{ где } e_2 = (\ldots, k_{a_J^r J}(a_J^i, \bar{k}_J)) \\ k_{(a_J^r)^\theta J}((a_J^i)^\theta, \bar{k}_J) = k_{BJ}((a_B^i)^\theta, (k_B^r)^\theta) \end{cases} \tag{39}$$

Из второй строчки в (39) следует, что

$$(a_J^r)^\theta = B, (a_J^i)^\theta = (a_B^i)^\theta, \bar{k}_J = (k_B^r)^\theta. \tag{40}$$

Из $\theta_2' \vdash at_J = 1$ следует, что $\exists\, \theta_3 \leq_\pi \theta_2'$:

$$\pi \ni J^{0,1} : \theta_3' \overset{\circ ? e_3}{\to} \theta_3, \text{ где } e_3 = (\ldots, k_{\hat{a}_J^r J}(\hat{a}_J^i, \hat{n}_J^i, \hat{n}_J^r)). \tag{41}$$

Из (40) и (41) следует, что $k_{BJ}(*,*,*) \subseteq e_3^\theta \in x_\circ^{\theta_3}$ (где звёздочки обозначают некоторые термы), откуда по теореме (2), с учетом соотношений $\theta_3 \models E \perp \dagger$ и $k_{BJ} \in E$ получаем: $\exists\, \theta_4 \leq_\pi \theta_3'$:

$$\begin{cases} \pi \ni R_{B_1}^{1,2} : \theta_4' \overset{\circ ! e_4}{\to} \theta_4, \text{ где } e_4 = (\ldots, k_{B_1 J}(a_{B_1}^i, n_{B_1}^i, \bar{n}_{B_1}^r)) \\ k_{B_1 J}((a_{B_1}^i)^\theta, (n_{B_1}^i)^\theta, \bar{n}_{B_1}^r) = k_{BJ}((a_B^i)^\theta, (n_J^i)^\theta, (n_J^r)^\theta) \end{cases} \tag{42}$$

Из второго равенства в (42) следует, что

$$B_1 = B, (n_B^i)^\theta = (n_J^i)^\theta, \bar{n}_B^r = (n_J^r)^\theta. \tag{43}$$

По теореме (2), из $\theta_1 \models E \perp \dagger$, $(k_B^r(n_B^r))^\theta \subseteq e_1^\theta \in x_\circ^{\theta_1}$, и $(k_B^r)^\theta = \bar{k}_J \in E$ следует, что $\exists\, \theta_5 \leq_\pi \theta_1'$:

$$\begin{cases} \pi \ni I_A^{2,3} : \theta_5' \overset{\circ ! e_5}{\to} \theta_5, \text{ где } e_5 = (\ldots, k_A^i(n_A^r)) \\ (k_A^i(n_A^r))^\theta = \bar{k}_J(n_B^r) \end{cases} \tag{44}$$

Из второй строчки в (44) следует, что

$$(k_A^i)^\theta = \bar{k}_J, (n_A^r)^\theta = n_B^r. \tag{45}$$

Из $\theta_5 \vdash at_{I_A} = 2$ следует, что $\exists\, \theta_6 \leq_\pi \theta_5'$:

$$\pi \ni I_A^{1,2} : \theta_6' \overset{\circ ? e_6}{\to} \theta_6, \text{ где } e_6 = (k_{AJ}(a_A^r, \hat{k}_A^i, n_A^i, \hat{n}_A^r), \ldots). \tag{46}$$

Из (45) и (46) следует, что

$$k_{AJ}(a^r_A,(k^i_A)^\theta,n^i_A,(n^r_A)^\theta)=k_{AJ}(a^r_A,\bar{k}_J,n^i_A,n^r_B)\subseteq e^\theta_6\in x^{\theta_6}_\circ. \tag{47}$$

По теореме (2), из $\theta_6\models E\perp\dagger$, $k_{AJ}\in E$, и (47) следует, что $\exists\,\theta_7\leq_\pi\theta'_6$:

$$\begin{cases}\pi\ni J^{1,2}_1:\theta'_7\overset{\circ!e_7}{\to}\theta_7,\ \text{где}\ e_7=(k_{a^i_{J_1}J_1}(a^r_{J_1},\bar{k}_{J_1},n^i_{J_1},n^r_{J_1}),\ldots)\\ k_{(a^i_{J_1})^\theta J}((a^r_{J_1})^\theta,\bar{k}_{J_1},(n^i_{J_1})^\theta,(n^r_{J_1})^\theta)=k_{AJ}(a^r_A,\bar{k}_J,n^i_A,n^r_B)\end{cases} \tag{48}$$

Из второй строчки в (48) следует, что

$$(a^i_{J_1})^\theta=A,(a^r_{J_1})^\theta=a^r_A,J_1=J,(n^i_J)^\theta=n^i_A,(n^r_J)^\theta=\bar{n}^r_B. \tag{49}$$

Свойство (38) следует из (40), (43), (45), (49). ■

Теорема 5 (аутентификация респондера перед инициатором)

РП (28) обладает следующим свойством: $\forall\,I_A\in\mathcal{P},\forall\,\theta\in\Theta_\mathcal{P}$, если $\theta\vdash at_{I_A}=2$, то $\exists\,R_B\in\mathcal{P}$:

$$\theta\vdash\left\{\begin{array}{l}at_{R_B}=2,\\ a^r_A=B,a^i_B=A,\\ n^i_A=n^i_B,n^r_A=n^r_B\end{array}\right\}. \tag{50}$$

Доказательство.

Пусть процесс I_A и состояние θ таковы, что $\theta\vdash at_{I_A}=2$, и $\pi=(\theta_0,\theta_1,\ldots)$ – выполнение РП $\mathcal{P}$, такое, что $\theta\in\pi$.

Из $\theta\vdash at_{I_A}=2$ следует, что $\exists\,\theta_1\leq_\pi\theta$:

$$\pi\ni I^{1,2}_A:\theta'_1\overset{\circ?e_1}{\to}\theta_1,\ \text{где}\ e_1=(k_{AJ}(a^r_A,\hat{k}^i_A,n^i_A,\hat{n}^r_A),\ldots). \tag{51}$$

По теореме 2, из $\theta_1\models E\perp\dagger$, $k_{AJ}(*,*,*,*)\subseteq e^\theta_1\in x^{\theta_1}_\circ$ (где звёздочки обозначают некоторые термы) и $k_{AJ}\in E$, следует, что $\exists\,\theta_2\leq_\pi\theta'_1$:

$$\begin{cases}\pi\ni J^{1,2}:\theta'_2\overset{\circ!e_2}{\to}\theta_2,\ \text{где}\ e_2=(k_{a^i_JJ}(a^r_J,\bar{k}_J,n^i_J,n^r_J),\ldots)\\ k_{(a^i_J)^\theta J}((a^r_J)^\theta,\bar{k}_J,(n^i_J)^\theta,(n^r_J)^\theta)=k_{AJ}(a^r_A,(k^i_A)^\theta,n^i_A,(n^r_A)^\theta)\end{cases} \tag{52}$$

Из второй строчки в (52) следует, что

$$\begin{aligned}&(a^i_J)^\theta=A,(a^r_J)^\theta=a^r_A,\bar{k}_J=(k^i_A)^\theta,\\ &(n^i_J)^\theta=n^i_A,(n^r_J)^\theta=(n^r_A)^\theta.\end{aligned} \tag{53}$$

Из $\theta_2 \vdash at_J = 1$ следует, что $\exists\, \theta_3 \leq_\pi \theta'_2$:

$$\pi \ni J^{0,1} : \theta'_3 \overset{\circ?e_3}{\to} \theta_3, \text{ где } e_3 = (\ldots, k_{\hat{a}^r_J J}(\hat{a}^i_J, \hat{n}^i_J, \hat{n}^r_J)). \tag{54}$$

Из (53) и (54) следует, что $k_{a^r_A J}(A, n^i_A, (n^r_A)^\theta) \subseteq e_3^\theta \in x_\circ^{\theta_3}$, откуда по теореме (2), учитывая $\theta_3 \models E \perp \dagger$, и $k_{a^r_A J} \in E$, получаем: $\exists\, \theta_4 \leq_\pi \theta'_3$:

$$\begin{cases} \pi \ni R_B^{1,2} : \theta'_4 \overset{\circ!e_4}{\to} \theta_4, \text{ где } e_4 = (\ldots, k_{BJ}(a^i_B, n^i_B, \bar{n}^r_B)) \\ k_{BJ}((a^i_B)^\theta, (n^i_B)^\theta, \bar{n}^r_B) = k_{a^r_A J}(A, n^i_A, (n^r_A)^\theta). \end{cases} \tag{55}$$

Второе равенство в (55) влечёт равенства, из которых следует (50):

$$B = a^r_A, (a^i_B)^\theta = A, (n^i_B)^\theta = n^i_A, \bar{n}^r_B = (n^r_A)^\theta. \quad \blacksquare$$

7. Верификация протокола передачи сообщений между несколькими агентами

В этом пункте рассматривается пример верификации КП, предназначенного для передачи ШС по открытому каналу между несколькими агентами. Данный КП является обобщением известного КП Wide-Mouth Frog и изложен в работе [96].

7.1. Описание протокола

Участники этого протокола – агенты из множества $Ag \subseteq Agents$ и доверенный посредник J. Каждый агент $A \in Ag$ использует для связи с J ключ k_{AJ}, доступный только A и J. Сеанс передачи сообщения x в зашифрованном виде от агента $A \in Ag$ агенту $B \in Ag$ включает в себя следующие действия:

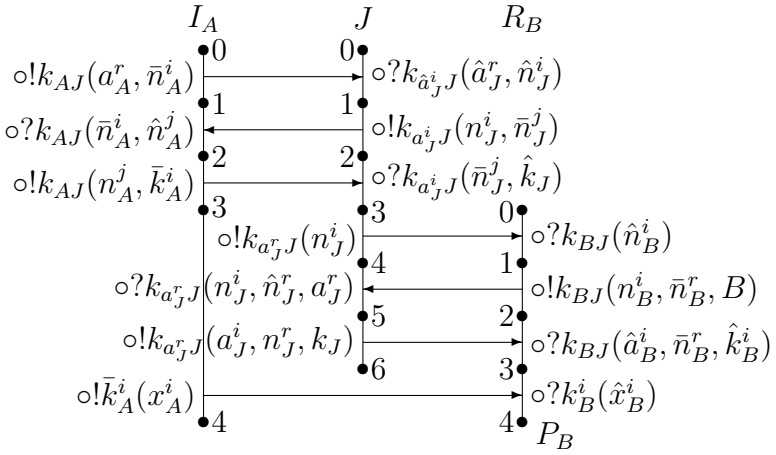
- обмен сообщениями между A и J, в результате чего J узнает имя A отправителя, имя B получателя, и ключ k, на котором будет зашифровано сообщение x от A для получателя B,
- обмен сообщениями между J и B, в результате чего B узнает имя A отправителя сообщения, которое B получит от A, и ключ k, на котором будет зашифровано это сообщение,
- пересылка ШС $k(x)$ от A к B.

Выполнение сеанса данного КП с инициатором A, респондером B и доверенным по-

средником J представляет собой следующую совокупность пересылок сообщений:

$$\begin{array}{lll} 1. & A \to J & : \ k_{AJ}(A, n_A) \\ 2. & J \to A & : \ k_{AJ}(n_A, n_J) \\ 3. & A \to J & : \ k_{AJ}(n_J, k) \\ 4. & J \to B & : \ k_{BJ}(n_A) \\ 5. & B \to J & : \ k_{BJ}(n_A, n_B, B) \\ 6. & J \to B & : \ k_{BJ}(A, n_B, k) \\ 7. & A \to B & : \ k(x) \end{array} \tag{56}$$

Данный сеанс представляется следующей схемой:



(57)

РП $\mathcal{P}$, соответствующий этому КП, имеет вид

$$\mathcal{P} = \{\{I_A^* \mid A \in Ag\}, \{R_B^* \mid B \in Ag\}, J^*, \dagger\}. \tag{58}$$

Свойства этого КП, которые должны быть верифицированы:

- **секретность** ключей, передаваемых сообщений и нонсов:

$$\forall\, c \in \Theta_{\mathcal{P}} \ \ \theta \models E \perp \dagger, \ \ \text{где } E = \{k_{AJ}, k_A^i, x_A^i, n_A^i \mid A \in Ag\} \tag{59}$$

- **целостность** передаваемых сообщений:

$$\begin{array}{l} \forall\, R_B \in \mathcal{P}, \forall\, \theta \in \Theta_{\mathcal{P}}, \text{если } \theta \vdash at_{R_B} = 4, \text{ то } \exists\ I_A \in \mathcal{P}: \\ \theta \vdash \{at_{I_A} = 4, a_A^r = B, a_B^i = A, n_A^i = n_B^i, k_A^i = k_B^i, x_A^i = x_B^i\} \end{array} \tag{60}$$

7.2. Верификация протокола

Доказательство свойства секретности (59) дословно повторяет начало рассуждений по доказательству аналогичного свойства протокола Yahalom.

Докажем свойство целостности (60). Будем использовать в этом доказательстве свойство (59) (не упоминая об этом).

Пусть процесс $R_B \in \mathcal{P}$ и состояние $\theta \in \Theta_{\mathcal{P}}$ таковы, что $\theta \vdash at_{R_B} = 4$. Докажем, что $\exists\, I_A \in \mathcal{P}$: выполнено утверждение во второй строчке (60).

Пусть π – путь из θ^0 в θ. Из $\theta \vdash at_{R_B} = 4$ следует, что

$$\begin{array}{l} \exists\, \theta_1 \leq_\pi \theta : \ \pi \ni R_B^{3,4} : \theta_1' \overset{\circ ? e_1}{\to} \theta_1, \text{ где } e_1 = k_B^i(\hat{x}_B^i) \\ \exists\, \theta_2 \leq_\pi \theta_1' : \ \pi \ni R_B^{2,3} : \theta_2' \overset{\circ ? e_2}{\to} \theta_2, \text{ где } e_2 = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i) \end{array} \tag{61}$$

По теореме 2, из второй строки в (61), $e_2^\theta \in x_\circ^{\theta_2}$, $k_{BJ} \in E$, следует:

$$\begin{cases} \exists\, \theta_3 \leq_\pi \theta_2' : \pi \ni J^{5,6} : \theta_3' \overset{\circ ! e_3}{\to} \theta_3, \text{ где } e_3 = k_{a_J^r J}(a_J^i, n_J^r, k_J) \\ k_{(a_J^r)^\theta J}((a_J^i)^\theta, (n_J^r)^\theta, (k_J)^\theta) = k_{BJ}(\hat{a}_B^i, \bar{n}_B^r, \hat{k}_B^i) \end{cases} \tag{62}$$

Из второй строки в (62) следует, что

$$(a_J^r)^\theta = B, \ (a_J^i)^\theta = (a_B^i)^\theta, \ (n_J^r)^\theta = \bar{n}_B^r, \ (k_J)^\theta = (k_B^i)^\theta. \tag{63}$$

Из первой строки в (62), с учетом (63), получаем:

$$\exists\, \theta_4 \leq_\pi \theta_3' : \pi \ni J^{4,5} : \theta_4' \overset{\circ ? e_4}{\to} \theta_4, \text{ где } e_4 = k_{BJ}(n_J^i, n_B^r, B). \tag{64}$$

По теореме 2, из (64), и того, что $e_4^\theta \in x_\circ^{\theta_4}$, $k_{BJ} \in E$, следует:

$$\begin{cases} \exists\, \theta_5 \leq_\pi \theta_4' : \pi \ni B_1^{1,2} : \theta_5' \overset{\circ ! e_5}{\to} \theta_5, \text{ где } e_5 = k_{B_1 J}(n_{B_1}^i, \bar{n}_{B_1}^r, B_1) \\ k_{B_1 J}((n_{B_1}^i)^\theta, \bar{n}_{B_1}^r, B_1) = k_{BJ}((n_J^i)^\theta, \bar{n}_B^r, B) \end{cases} \tag{65}$$

Из второй строки в (65) следует, что

$$\bar{n}_{B_1}^r = \bar{n}_B^r, \ B_1 = B, \ (n_B^i)^\theta = (n_J^i)^\theta. \tag{66}$$

Из (64) следует, что

$$\exists\, \theta_6 \leq_\pi \theta_4' : \pi \ni J^{2,3} : \theta_6' \overset{\circ ? e_6}{\to} \theta_6, \text{ где } e_6 = k_{a_J^i J}(n_J^j, k_J). \tag{67}$$

По теореме 2, из (67), и того, что $e_6^\theta \in x_\circ^{\theta_6}$, $k_{(a_J^i)^\theta J} \in E$, следует:

$$\begin{cases} \exists\, \theta_7 \leq_\pi \theta_6' : \pi \ni A^{2,3} : \theta_7' \overset{\circ ! e_7}{\to} \theta_7, \text{ где } e_7 = k_{AJ}(n_A^j, \bar{k}_A^i) \\ k_{AJ}((n_A^j)^\theta, \bar{k}_A^i) = k_{(a_J^i)^\theta J}(\bar{n}_J^j, (k_J)^\theta) \end{cases} \tag{68}$$

Из второй строки в (68) получаем:

$$A = (a_J^i)^\theta, \ (n_A^j)^\theta = \bar{n}_J^j, \ \bar{k}_A^i = (k_J)^\theta \tag{69}$$

Из первой строки в (68) получаем:

$$\exists\,\theta_8 \le_\pi \theta'_7 : \pi \ni I_A^{1,2} : \theta'_8 \overset{\circ?e_8}{\to} \theta_8,\ \text{где}\ e_8 = k_{AJ}(\bar{n}^i_A, \hat{n}^j_A). \tag{70}$$

По теореме 2, из (70), и того, что $e_8^\theta \in x_\circ^{\theta_8}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\,\theta_9 \le_\pi \theta'_8 : \pi \ni {J_1}^{1,2} : \theta'_9 \overset{\circ!e_9}{\to} \theta_9,\ \text{где}\ e_9 = k_{a^i_{J_1}J}(n^i_{J_1}, \bar{n}^j_{J_1}) \\ k_{(a^i_{J_1})^\theta J}((n^i_{J_1})^\theta, \bar{n}^j_{J_1}) = k_{AJ}(\bar{n}^i_A, (n^j_A)^\theta) \end{cases} \tag{71}$$

Из второй строки в (71) получаем:

$$(a^i_{J_1})^\theta = A,\ (n^i_{J_1})^\theta = \bar{n}^i_A,\ \bar{n}^j_{J_1} = (n^j_A)^\theta \tag{72}$$

Из (69) и (72) получаем:

$$\bar{n}^j_{J_1} = \bar{n}^j_J = (n^j_A)^\theta,\ J_1 = J,\ (a^i_J)^\theta = A,\ (n^i_J)^\theta = \bar{n}^i_A. \tag{73}$$

Из (63) и (69) получаем:

$$(k^i_B)^\theta = (k_J)^\theta = \bar{k}^i_A \in E, \tag{74}$$

поэтому по теореме 2, из первой строки в (61) и $e_1^\theta \in x_\circ^{\theta_1}$ следует:

$$\begin{cases} \exists\,\theta_{11} \le_\pi \theta'_1 : \pi \ni {A_1}^{3,4} : \theta'_{11} \overset{\circ!e_{11}}{\to} \theta_{11},\ \text{где}\ e_{11} = \bar{k}^i_{A_1}(x^i_{A_1}) \\ \bar{k}^i_{A_1}(x^i_{A_1}) = (k^i_B)^\theta((x^i_B)^\theta) \end{cases} \tag{75}$$

Из второй строки в (75) и (74) получаем:

$$\bar{k}^i_{A_1} = (k^i_B)^\theta = \bar{k}^i_A,\ A_1 = A,\ x^i_A = (x^i_B)^\theta. \tag{76}$$

Из первой строки в (71) и (73) получаем:

$$\exists\,\theta_{10} \le_\pi \theta'_9 : \pi \ni J^{0,1} : \theta'_{10} \overset{\circ?e_{10}}{\to} \theta_{10},\ \text{где}\ e_{10} = k_{\hat{a}^i_J J}(\hat{a}^r_J, \hat{n}^i_J). \tag{77}$$

Из (63) и (73) следует, что $e_{10}^\theta = k_{AJ}(B, \bar{n}^i_A)$.

По теореме 2, из (77), $e_{10}^\theta \in x_\circ^{\theta_{10}}$, $k_{AJ} \in E$, следует:

$$\begin{cases} \exists\,\theta_{12} \le_\pi \theta'_{10} : \pi \ni {A_1}^{0,1} : \theta'_{12} \overset{\circ!e_{12}}{\to} \theta_{12},\ \text{где}\ e_{12} = k_{A_1J}(a^r_{A_1}, \bar{n}^i_{A_1}) \\ k_{A_1J}(a^r_{A_1}, \bar{n}^i_{A_1}) = k_{AJ}(B, \bar{n}^i_A) \end{cases} \tag{78}$$

Из второй строки в (78) получаем:

$$\bar{n}^i_{A_1} = \bar{n}^i_A,\ A_1 = A,\ a^r_A = B. \tag{79}$$

Утверждение (60) обосновывается следующим образом:

- $\theta \vdash at_{I_A} = 4$ следует из (75), (76): $\theta_{11} \vdash at_{A_1} = 4$, $A_1 = A$, $\theta_{11} \leq_\pi s$,
- $\theta \vdash a^r_A = B$ следует из (79),
- $\theta \vdash a^i_B = A$ следует из (63) и (69),
- $\theta \vdash n^i_A = n^i_B$ следует из (66) и (73),
- $\theta \vdash k^i_A = k^i_B$ следует из (74),
- $\theta \vdash x^i_A = x^i_B$ следует из (76). ■

8. Заключение

В настоящей работе была построена новая модель КП, и показаны примеры ее использования для решения задач верификации свойств секретности и соответствия.

Для дальнейшей деятельности по развитию данной модели и основанных на ней методов верификации можно назвать следующие задачи:

- развитие языков спецификаций свойств КП, позволяющих выражать например свойства нулевого разглашения в КП аутентификации, свойства неотслеживаемости в КП электронных платежей, свойства анонимности и правильности подсчета голосов в КП электронного голосования, и разработка методов верификации свойств, выражаемых на этих языках,
- построение методов автоматизированного синтеза КП по описанию свойств, которым они должны удовлетворять.

Список литературы

1. Кондратьев Д.А., Бояндин Л.К., Гончар Г.Е., Марченко В.В., Обухова А.А., Разбитнова Ю.Ю., Хованская А.С., Янбулатов Д.Р. Формальная верификация реализации хэш-функции «Стрибог» с «Группой Астра», Системная информатика, 2025. – № 28, – С. 25-52.
2. Миронов А.М., Математическая модель и методы верификации криптографических протоколов, Интеллектуальные системы. Теория и приложения, издательство ООО "Интеллектуальные системы"(Москва), том 26, № 2, с. 85-144, 2022.
3. Миронов А.М. Методы верификации программ. ДМК Пресс Москва, 2023, 335 с.
4. M. Abadi and B. Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. In Journal of the ACM, 52(1), pp. 102-146, 2005.
5. M. Abadi, B. Blanchet, C. Fournet. Just Fast Keying in the Pi Calculus. In ACM

Transactions on Information and System Security, 10(3), 2007.

6. M. Abadi, B. Blanchet, C. Fournet. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. Journal of the ACM, Volume 65, Issue 1 Article No.: 1, Pages 1 - 41, 2017.
7. M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: the Spi calculus. Inf. Comput. 148, issue 1, 1–70 (1999)
8. M. Abadi, C. Fournet, Mobile values, new names, and secure communication, in 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'01), ed. by C. Hankin, D. Schmidt, London, UK (ACM, New York, 2001), pp. 104–115
9. M. Abadi, M.R. Tuttle, A semantics for a logic of authentication (extended abstract), in 10th ACM Symposium on Principles of Distributed Computing (PODC'91), Montreal, Canada (ACM, New York, 1991), pp. 201–216
10. R. Anderson and R. Needham. Programming Satan's computer. In J. van Leeuwen (ed.) Computer Science Today, volume 1000 of LNCS. Springer, 1995, pp. 426–440.
11. Andrew W. Appel, Verification of a Cryptographic Primitive: SHA-256, ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 37, Issue 2, p. 1-31, 2015.
12. G. Bella. Inductive Verification of Cryptographic Protocols. PhD thesis, Cambridge University, 2000.
13. B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In 14th IEEE Computer Security Foundations Workshop (CSFW), pp. 82-96, 2001.
14. Bruno Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, Foundations and Trends in Privacy and Security: Vol. 1: No. 1-2, pp 1-135, 2016.
15. Burrows M., Abadi M., Needham R., A Logic of Authentication. In ACM Transactions on Computer Systems, 8(1), (1990) 18-36.
16. I. Cervesato, N.A. Durgin, P.D. Lincoln, J.C. Mitchell, A. Scedrov. A Comparison between Strand Spaces and Multiset Rewriting for Security Protocol Analysis. Journal of Computer Security, vol. 13, no. 2, pp. 265-316, 2005
17. Cervesato I., Jaggard A.D., Scedrov A., Tsay J.-K., Walstad C., Breaking and fixing public-key Kerberos, Information and Computation Volume 206, Issues 2-4, (2008), Pages 402-424.
18. Véronique Cortier, Stephanie Delaune, and Vaishnavi Sundararajan. A Decidable Class of Security Protocols for Both Reachability and Equivalence Properties. Journal of

Automated Reasoning, vol. 65, issue 4, 479–520, April 2021.

19. V. Cortier and S. Kremer, editors. Formal Models and Techniques for Analyzing Security Protocols, volume 5 of Cryptology and Information Security Series. IOS Press, 2011.
20. Véronique Cortier, Steve Kremer. Formal Models and Techniques for Analyzing Security Protocols: A Tutorial. Foundations and Trends in Programming Languages, 1(3):151–267, (2014)
21. Véronique Cortier and Cyrille Wiedling. A formal analysis of the Norwegian E-voting protocol. Journal of Computer Security, 25(1):21–57, 2016.
22. C.J.F. Cremers, On the protocol composition logic PCL, in ACM Symposium on Information, Computer & Communication Security (ASIACCS'08), ed. by M. Abe, V. Gligor, Tokyo, Japan (ACM, New York, 2008), pp. 66–76.
23. Cas Cremers, Sjouke Mauw. Operational Semantics and Verification of Security Protocols, Springer-Verlag Berlin Heidelberg, 2012.
24. A. Datta, A. Derek, J.C. Mitchell, A. Roy, Protocol Composition Logic (PCL), in Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin, ed. by L. Cardelli, M. Fiore, G. Winskel. Electronic Notes in Theoretical Computer Science, vol. 172, (2007), pp. 311– 358.
25. A. Datta, J.C. Mitchell, A. Roy, S.H. Stiller, Protocol composition logic, in Formal Models and Techniques for Analyzing Security Protocols, ed. by V. Cortier, S. Kremer (IOS Press, Lansdale, 2011)
26. Denning D.E., Sacco G., Timestamps in Key Distribution Protocols, Communications of the ACM, Vol. 24, No. 8, (1981) 533-536.
27. S.F. Doghmi, J.D. Guttman, F.J. Thayer, Searching for shapes in cryptographic protocols, in 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07), ed. by O. Grumberg, M. Huth, Braga, Portugal. Lecture Notes in Computer Science, vol. 4424 (Springer, Berlin, 2007), pp. 523–537
28. S. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons and the shapes of bundles, in 7th International Workshop on Issues in the Theory of Security (WITS'07), Braga, Portugal (2007)
29. S.F. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons, homomorphisms, and shapes: characterizing protocol executions, in 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII), New Orleans, USA. Electronic Notes in

Theoretical Computer Science, vol. 173 (Elsevier, Amsterdam, 2007), pp. 85–102

30. N.A. Durgin, J.C. Mitchell, D. Pavlovic, A compositional logic for protocol correctness, in 14th IEEE Computer Security Foundations Workshop (CSFW'01), Cape Breton, Canada (IEEE Computer Society, Los Alamitos, 2001), pp. 241–255.
31. B. Dutertre and S. A. Schneider. Using a PVS embedding of CSP to verify authentication protocols. Number 1275 in LNCS. Springer, 1997, pp. 121-136.
32. L. Gong, R.M. Needham, R. Yahalom, Reasoning about belief in cryptographic protocols, in 1990 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, USA (IEEE Computer Society, Los Alamitos, 1990), pp. 234–248
33. Joshua D. Guttman. State and Progress in Strand Spaces: Proving Fair Exchange. Journal of Automated Reasoning, 48(2): 159-195, 2012.
34. J. D. Guttman and F. J. Thayer. Authentication tests. IEEE Computer Society Symposium on Research in Security and Privacy, 2000, pp. 96-109.
35. J.D. Guttman, F.J. Thayer, Authentication tests and the structure of bundles. Theor. Comput. Sci. 283(2), 333–380 (2002)
36. C. A. R. Hoare. Communicating Sequential Processes. Prentice-Hall, 1985, 256 pages.
37. R.A. Kemmerer, C. Meadows, J.K. Millen, Three systems for cryptographic protocol analysis. J. Cryptol. 7, 79–130 (1994)
38. S. Kremer, M. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In 14th European Symposium on Programming (ESOP), pp. 186-200, 2005.
39. Yongjian Li, Jun Pang. An inductive approach to strand spaces. Formal Aspects of Computing, Vol. 25, No. 4, 2013, pp. 465–501.
40. Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131–133, November 1995.
41. G. Lowe and B. Roscoe. Using CSP to detect errors in the TMN protocol. IEEE Transactions in Software Engineering, 23(10), 1997, pp. 659 - 669.
42. R. Milner, A Calculus of Communicating Systems, Springer Verlag, 1980, 176 p.
43. Needham R.M., Schroeder M.D., Authentication revisited, ACM SIGOPS Operating Systems Review, Vol. 21, No. 1, p. 7 (1987).
44. Roger Needham and Michael Schroeder. Using encryption for authentification in large networks of computers. Communications of the ACM, 21(12), December 1978, pp. 993-999.

45. P.C. van Oorschot, Extending cryptographic logics of belief to key agreement protocols, in 1st ACM Conference on Computer and Communications Security (ACM CCS'93), ed. by D.E. Denning, R. Pyle, R. Ganesan, R.S. Sandhu, V. Ashby, Fairfax, USA (ACM, New York, 1993), pp. 232–243
46. L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. In ACM Trans. on Information and System Security, 2(3), pp. 332-351, 1999.
47. L.C. Paulson, The inductive approach to verifying cryptographic protocols. J. Comput. Secur. 6(1–2), 85–128 (1998)
48. L.C. Paulson, Proving properties of security protocols by induction, in 10th IEEE Computer Security Foundations Workshop (CSFW'97), Rockport, Massachusetts (IEEE Computer Society, Los Alamitos, 1997), pp. 70–83.
49. Roggenbach, M., Cerone, A., Schlingloff, B.- H., Schneider, G., Shaikh, S.A., Formal verification of security protocols, in: Formal Methods for Software Engineering: Languages, Methods, Application Domains (Texts in Theoretical Computer Science. An EATCS Series) 1st ed., Springer International Publishing, 2022.
50. P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. Journal of Computer Security, vol. 9, issue 1-2, 2001, pp. 75-103.
51. P.Y.A. Ryan, S.A. Schneider, M.H. Goldsmith, G. Lowe and A.W. Roscoe. The Modelling and Analysis of Security Protocols: the CSP Approach, Addison-Wesley, 2000, 320 pages.
52. Mark D. Ryan and Ben Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Edited by Véronique Cortier and Steve Kremer, 2011 IOS Press, p. 112-142.
53. S. A. Schneider. Verifying authentication protocols in CSP. IEEE Transactions on Software Engineering, vol. 24, issue 9, 1998, pp. 741-758.
54. S. Schneider, Security properties and CSP, in 17th IEEE Symposium on Security & Privacy (S&P'96), Oakland, USA (IEEE Computer Society, Los Alamitos, 1996), pp. 174–187.
55. S. A. Schneider and A. Sidiropoulos. CSP and anonymity. ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security: Computer Security Pages 198 - 218
56. S.G. Stubblebine, R.N. Wright, An authentication logic with formal semantics supporting synchronization, revocation, and recency. IEEE Trans. Softw. Eng. 28(3), 256–285 (2002)
57. Syverson P., Meadows C., A Logical Language for Specifying Cryptographic Protocol

Requirements, Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy, (1993) 165-177, IEEE Computer Society Press.

58. P.F. Syverson, P.C. van Oorschot, A unified cryptographic protocol logic. CHACS Report 5540-227 NRL (1996)
59. F.J. Thayer Fábrega, J.C. Herzog, J.D. Guttman, Honest ideals on Strand Spaces, in 11th IEEE Computer Security Foundations Workshop (CSFW'98), Rockport, USA (IEEE Computer Society, Los Alamitos, 1998), pp. 66–77
60. F.J. Thayer Fábrega, J.C. Herzog, J.D. Guttman, Mixed Strand Spaces, in 12th IEEE Computer Security Foundations Workshop (CSFW'99), IEEE Computer Society, Los Alamitos, 1999, pp. 72–82
61. F.J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. Journal of Computer Security, 7(2/3):191-230, 1999.
62. F.J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces: why is a security protocol correct? 1998 IEEE Symposium on Security and Privacy, 1998, pp. 160-171.
63. Fan Yang, Santiago Escobar, Catherine A Meadows, José Meseguer. Strand Spaces with Choice via a Process Algebra Semantics. PPDP '16: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, September 2016, pages 76–89.
64. Véronique Cortier, Stephanie Delaune, and Vaishnavi Sundararajan. A Decidable Class of Security Protocols for Both Reachability and Equivalence Properties. Journal of Automated Reasoning, 65:479–520, April 2021.
65. Roggenbach, M., Cerone, A., Schlingloff, H., Schneider, G., Shaikh, S.A., Formal verification of security protocols, in: Formal Methods for Software Engineering: Languages, Methods, Application Domains (Texts in Theoretical Computer Science. An EATCS Series) 1st ed., Springer International Publishing, 2021.
66. Véronique Cortier and Cyrille Wiedling. A formal analysis of the Norwegian E-voting protocol. Journal of Computer Security, 25(15777):21–57, 2017.
67. M. Abadi, B. Blanchet, C. Fournet. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. [Research Report] ArXiv. 2016, pp.110. `hal-01423924`, `https://arxiv.org/abs/1609.03003`
68. Bruno Blanchet, Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif, 2016.

69. Fan Yang, Santiago Escobar, Catherine A Meadows, José Meseguer. Strand Spaces with Choice via a Process Algebra Semantics. PPDP '16: Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, September 2016, pages 76–89.
70. Véronique Cortier, Steve Kremer. Formal Models and Techniques for Analyzing Security Protocols: A Tutorial. Foundations and Trends in Programming Languages, 1(3):151–267, (2014)
71. Yongjian Li, Jun Pang. An inductive approach to strand spaces. Formal Aspects of Computing, Vol. 25, No. 4, 2013.
72. Cas Cremers, Sjouke Mauw. Operational Semantics and Verification of Security Protocols, Springer-Verlag Berlin Heidelberg, 2012.
73. Joshua D. Guttman. State and Progress in Strand Spaces: Proving Fair Exchange. Journal of Automated Reasoning, 48(2): 159-195, 2012.
74. V. Cortier and S. Kremer, editors. Formal Models and Techniques for Analyzing Security Protocols, volume 5 of Cryptology and Information Security Series. IOS Press, 2011.
75. A. Datta, J.C. Mitchell, A. Roy, S. Stiller, Protocol composition logic, in Formal Models and Techniques for Analyzing Security Protocols, ed. by V. Cortier, S. Kremer (IOS Press, Lansdale, 2011)
76. Mark D. Ryan and Ben Smyth, Applied pi calculus, in: Formal Models and Techniques for Analyzing Security Protocols, Edited by Véronique Cortier, 2011 IOS Press, p. 112-142.
77. C.J.F. Cremers, On the protocol composition logic PCL, in ACM Symposium on Information, Computer & Communication Security (ASIACCS'08), ed. by M. Abe, V. Gligor, Tokyo, Japan (ACM, New York, 2008), pp. 66–76
78. Cervesato I., Jaggard A.D., Scedrov A., Tsay J.-K., Walstad C., Breaking and fixing public-key Kerberos, Information and Computation Volume 206, Issues 2-4, (2008), Pages 402-424.
79. M. Abadi, B. Blanchet, C. Fournet, Just Fast Keying in the Pi Calculus. In ACM Transactions on Information and System Security, 10(3), 2007.
80. A. Datta, A. Derek, J.C. Mitchell, A. Roy, Protocol Composition Logic (PCL), in Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin, ed. by L. Cardelli, M. Fiore, G. Winskel. Electronic Notes in Theoretical Computer Science, vol. 172, (2007), pp. 311– 358
81. S. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons and the shapes of bundles, in 7th

International Workshop on Issues in the Theory of Security (WITS'07), Braga, Portugal (2007)

82. S.F. Doghmi, J.D. Guttman, F.J. Thayer, Skeletons, homomorphisms, and shapes: characterizing protocol executions, in 23rd Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIII), New Orleans, USA. Electronic Notes in Theoretical Computer Science, vol. 173 (Elsevier, Amsterdam, 2007), pp. 85–102
83. S.F. Doghmi, J.D. Guttman, F.J. Thayer, Searching for shapes in cryptographic protocols, in 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'07), ed. by O. Grumberg, M. Huth, Braga, Portugal. Lecture Notes in Computer Science, vol. 4424 (Springer, Berlin, 2007), pp. 523–537
84. M. Abadi and B. Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. In Journal of the ACM, 52(1), pp. 102-146, 2005.
85. I. Cervesato, N.A. Durgin, P.D. Lincoln, J.C. Mitchell, A. Scedrov. A Comparison between Strand Spaces and Multiset Rewriting for Security Protocol Analysis. Journal of Computer Security, vol. 13, no. 2, pp. 265-316, 2005
86. S. Kremer, M. Ryan. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. In 14th European Symposium on Programming (ESOP), pp. 186-200, 2005.
87. J.D. Guttman, F.J. Thayer, Authentication tests and the structure of bundles. Theor. Comput. Sci. 283(2), 333–380 (2002)
88. S.G. Stubblebine, R.N. Wright, An authentication logic with formal semantics supporting synchronization, revocation, and recency. IEEE Trans. Softw. Eng. 28(3), 256–285 (2002)
89. M. Abadi, C. Fournet, Mobile values, new names, and secure communication, in 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'01), ed. by C. Hankin, D. Schmidt, London, UK (ACM, New York, 2001), pp. 104–115
90. B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In 14th IEEE Computer Security Foundations Workshop (CSFW), pp. 82-96, 2001.
91. N.A. Durgin, J.C. Mitchell, D. Pavlovic, A compositional logic for protocol correctness, in 14th IEEE Computer Security Foundations Workshop (CSFW'01), Cape Breton, Canada (IEEE Computer Society, Los Alamitos, 2001), pp. 241–272
92. G. Bella. Inductive Verification of Cryptographic Protocols. PhD thesis, Cambridge University, 2000.
93. J. D. Guttman and F. J. Thayer. Authentication tests and the normal, efficient penetrator.

IEEE Computer Society Symposium on Research in Security and Privacy, 2000.

94. P.Y.A. Ryan, S.A. Schneider, M.H. Goldsmith, G. Lowe and A.W. Roscoe. The Modelling and Analysis of Security Protocols: the CSP Approach, Addison-Wesley, 2000.
95. P. Y. A. Ryan and S. A. Schneider. Process algebra and non-interference. Journal of Computer Security, 2000.
96. M. Abadi, A.D. Gordon, A calculus for cryptographic protocols: the Spi calculus. Inf. Comput. 148, 1–70 (1999)
97. L. C. Paulson. Inductive Analysis of the Internet Protocol TLS. In ACM Trans. on Information and System Security, 2(3), pp. 332-351, 1999.
98. F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: Proving security protocols correct. Journal of Computer Security, 7(2/3):191-230, 1999.
99. F.J. Thayer, J.C. Herzog, J.D. Guttman, Mixed Strand Spaces, in 12th IEEE Computer Security Foundations Workshop (CSFW'99), IEEE Computer Society, Los Alamitos, 1999, pp. 72–82
100. L.C. Paulson, The inductive approach to verifying cryptographic protocols. J. Comput. Secur. 6(1–2), 85–128 (1998)
101. F.J. Thayer, J.C. Herzog, J.D. Guttman, Honest ideals on Strand Spaces, in 11th IEEE Computer Security Foundations Workshop (CSFW'98), Rockport, USA (IEEE Computer Society, Los Alamitos, 1998), pp. 66–77
102. F. J. Thayer, J. C. Herzog, and J. D. Guttman. Strand spaces: why is a security protocol correct? IEEE Computer Society Symposium on Security and Privacy, 1998.
103. S. A. Schneider. Verifying authentication protocols in CSP. IEEE Transactions on Software Engineering, 1998.
104. B. Dutertre and S. A. Schneider. Embedding CSP in PVS. An application to authentication protocols. Theorem proving in Higher Order Logics, number 1275 in LNCS. Springer, 1997.
105. G. Lowe and A. W. Roscoe. Using CSP to detect errors in the TMN protocol. IEEE Transactions in Software Engineering, 23(10), 1997.
106. L.C. Paulson, Proving properties of security protocols by induction, in 10th IEEE Computer Security Foundations Workshop (CSFW'97), Rockport, Massachusetts (IEEE Computer Society, Los Alamitos, 1997), pp. 70–83
107. S. Schneider, Security properties and CSP, in 17th IEEE Symposium on Security & Privacy (S&P'96), Oakland, USA (IEEE Computer Society, Los Alamitos, 1996), pp. 174–187.

108. P.F. Syverson, P.C. van Oorschot, A unified cryptographic protocol logic. CHACS Report 5540-227 NRL (1996)
109. S. A. Schneider and A. Sidiropoulos. CSP and anonymity. European Symposium on Research in Computer Security, 1996.
110. R. Anderson and R. Needham. Programming Satan's computer. In J. van Leeuwen (ed.) Computer Science Today, volume 1000 of LNCS. Springer, 1995.
111. Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131–136, November 1995.
112. R.A. Kemmerer, C. Meadows, J.K. Millen, Three systems for cryptographic protocol analysis. J. Cryptol. 7, 79–130 (1994)
113. P.C. van Oorschot, Extending cryptographic logics of belief to key agreement protocols, in 1st ACM Conference on Computer and Communications Security (ACM CCS'93), ed. by D.E. Denning, R. Pyle, R. Ganesan, R.S. Sandhu, V. Ashby, Fairfax, USA (ACM, New York, 1993), pp. 232–243
114. Syverson P., Meadows C., A Logical Language for Specifying Cryptographic Protocol Requirements, Proceedings of the 1993 IEEE Computer Security Symposium on Security and Privacy, (1993) 165-177, IEEE Computer Society Press.
115. M. Abadi, M. Tuttle, A semantics for a logic of authentication, in 10th ACM Symposium on Principles of Distributed Computing (PODC'91), Montreal, Canada (ACM, New York, 1991), pp. 201–216
116. Burrows M., Abadi M., Needham R., A Logic of Authentication. In ACM Transactions on Computer Systems, 8(1), (1990) 18-36.
117. L. Gong, R.M. Needham, R. Yahalom, Reasoning about belief in cryptographic protocol analysis, in 11th IEEE Symposium on Security & Privacy (S&P'90), Oakland, USA (IEEE Computer Society, Los Alamitos, 1990), pp. 234–248
118. Needham R., Schroeder M., Authentication revisited, Operating Systems Review, Vol. 21, No. 1, (1987).
119. C. A. R. Hoare. Communicating Sequential Processes. Prentice-Hall, 1985.
120. Denning D., Sacco G., Timestamps in Key Distribution Protocols, Communications of the ACM, Vol. 24, No. 8, (1981) 533-536.
121. R. Milner, A Calculus of Communicating Systems, Springer Verlag, 1980.
122. Roger Needham and Michael Schroeder. Using encryption for authentification in large

networks of computers. Communications of the ACM, 21(12), December 1978.

123. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In POPL'01: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pages 104–115. ACM Press, 2001.
124. M. Abadi, B. Blanchet, and C. Fournet. 2017. The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication. J. ACM 65, 1, Article 1 (October 2017), 103 pages.
125. Миронов А.М. Методы верификации программ. ДМК Пресс Москва, 2023, 335 с.

УДК 004.3; 378; 372.8

Опыт по исследованию эффективности учебного процесса при поддержке техническими средствами

Тихонова Т.И. (Институт систем информатики СО РАН, Новосибирский государственный университет)

Насибулов И.А. (Институт систем информатики СО РАН, Новосибирский государственный университет)

В данной работе описан опыт применения автоматизированной системы тестирования для поддержки учебного процесса на уроках информатики и ИКТ. Система используется для аккумулирования материалов по определенным тематическим разделам и проверки правильности решения на основе подобранных тестов. Система тестов создается с учетом отслеживания эффективности выбранных при реализации алгоритмов для подобранных по теме задач. Задействование данного механизма показало хорошую результативность и эффективность как в процессе подготовки к соревнованиям олимпиадного характера, так и в регулярных занятиях. Приведен анализ опыта использования и обозначены перспективы реализации возможностей автоматизированной системы для сопровождения учебного процесса. Технологии проверки заданий, поддержанные техническими средствами, перспективное и действенное направление. Материал может быть полезен как преподавателям школ, так и вузов для применения в образовательном процессе, а также для самостоятельной подготовки обучающихся.

Ключевые слова: *технические средства, система тестирования, обучение, информатика, методика, контроль и оценка, академическая обратная связь, рефлексия, индивидуальный подход.*

1. Введение

Поиск эффективных и результативных инструментов для повышения качества образования постоянно стоит на повестке дня. В научной теории процесс обучения содержит разработку приемов и способов организации познавательной деятельности, которые обеспечивают эффективное усвоение ими знаний, выработку умений и навыков и формирование разноплановых способностей. Бенджамин Блум [1] занимался иерархией обучения. По его методике можно рассматривать эффективность процесса обучения от простого запоминания фактов и информации до высочайшего (сложные мыслительные

процессы, включающие оценку и суждения о ценности материалов, которые им выданы в образовательном процессе).

Иерархия по Блуму [2] включала шесть уровней. Базовый – «знание» («понимание»). На этом уровне ученики запоминают и воспроизводят изученный материал. Уровень считается достигнутым, если обучающийся воспроизводит термины, конкретные факты, основные понятия, принципы и т. д.

За базовым следовали операции и связанные с ним навыки — понимание, применение, анализ, синтез и оценка. Каждый уровень включал в себя ряд когнитивных действий, которые он предполагает, сформулированных в виде глаголов. Каждый уровень представляет собой этап в развитии когнитивных способностей. Действия становятся более сложными по мере продвижения по уровням. Подробнее раскроем содержание каждого уровня;

- **Запоминание** — это первый уровень, который включает в себя извлечение, распознавание и припоминание соответствующих знаний из долговременной памяти. Действия на этом этапе могут включать запоминание фактов, припоминание информации или перечисление основных концепций.
- **Понимание** — это второй уровень, сосредоточенный на интерпретации, обобщении и объяснении содержания. Обучающимся на этом уровне может быть предложено перефразировать информацию, описать отношения между идеями или сравнить и противопоставить концепции.
- Третий уровень – **применение –** требует от учеников использовать свои знания и понимание концепции в новых ситуациях или контекстах. Это может включать решение проблем, демонстрацию методов или использование информации в реальных жизненных ситуациях.
- На четвертом уровне **(Анализ)** должно разбить информацию на ее составные части и определить, как они связаны. Этот уровень включает сравнение и сопоставление идей, изучение закономерностей и выявление базовых структур или отношений.
- Пятый уровень, **Оценка**, включает в себя вынесение суждений на основе данных, основанных на критериях и стандартах. На этом этапе ожидается, что обучающиеся будут оценивать информацию, критиковать теории или оценивать эффективность подходов, всегда полагаясь на доказательства для поддержки своих суждений.
- Последний уровень, **Творчество**, представляет собой наивысшее когнитивное развитие согласно таксономии Блума. На этом этапе требуется синтезировать информацию из

нескольких источников, генерировать новые идеи или разрабатывать оригинальные решения проблем.

Рис. 1. Иерархия по Блуму

Опираясь на данную иерархию, в информатике, безусловно, можно разрабатывать подходы индивидуально, ориентируясь на соответствующие способности учащихся. Кроме того, продвижение по уровням «пирамиды» обеспечивает комплексный опыт обучения и поддерживает продвижение по различным этапам когнитивного развития.

Осваивая каждый уровень, возможно получить более глубокое понимание предмета, развивая при этом критическое мышление, навыки решение проблем, адаптацию к текущим запросам, необходимые для академического и профессионального успеха.

Очень важный итог обучения – формирование человека творческого, умеющего не только использовать шаблонные решения, но стремящегося найти новые направления, алгоритмы решений, совместить подходы из различных областей знаний. Безусловно, творческий подход не всегда востребован. Регулярные задачи, которые решаются ежедневно, весьма требовательны к четкости, использованию надежных методов, прошедших апробацию. Но тем ценнее бывают рационализаторские и научные находки, которые творческий человек осуществит, а другим нужно научиться пользоваться для улучшения создания продукта.

Работа в команде по созданию проектов [3] была неоднократно подтверждена как эффективная методика прохождения всех уровней, от первого до достижения наивысших (в соответствии с теорией Блума). Методика описана в работах по Летним школам юных

программистов и прочим апробациям метода проектов в малых группах [4], в том числе со студентами и школьниками [5, 6, 7, 8].

Все участники образовательного процесса заинтересованы в комфортности взаимодействия. Демократический стиль преподавательского (педагогического) общения в качестве средства управления активностью обучающихся, направленный на формирование и развитие личности, весьма эффективен. Движение к цели развития личностных, социальных, интеллектуальных и физических качеств обучающихся, основанный на действующем законодательстве, не отменяет качественные показатели, результативность и осмысление тематического плана. Зачастую результативность и качество процесса обучения страдают по причине различных факторов. В перечне таковых можно отметить момент отслеживания успешного осваивания учебного материала, который является существенным в осознанном использовании дополнительных заданий для отработки пройденного материала. Индивидуальный подход к разным по уровню подготовки обучающимся значимо влияет на атмосферу в коллективе, на продвижение в предметной области как продвинутых и заинтересованных, так и медленно включающихся в процесс обучения, особенно тех, кому необходимо большее время для осознанной отработки моментов теоретических или технических. Категория часто отсутствующих на регулярных занятиях по причине болезни или участия успешных учащихся в различных мероприятиях олимпиадного и соревновательного характера, одинаково отрицательно сказываются на процессе обучения. Для первых существенным является возможность отработки учебного материала в неспешном режиме, а вот для одаренных детей зачастую проблема бывает в ином. Они быстро усваивают материал, но не утруждают себя решением простых на их взгляд заданий, приступают к сложным. На первый взгляд, результат вполне хорош. Но так ли замечательно компенсируется взаимодействие преподавателя и ученика? Участие в проблемном обсуждении и активном обмене идеями и мнениями на уроках способствует развитию критического мышления, умению анализировать информацию, слушать и уважать точки зрения других людей, а также совместному поиску знаний и решений. Также при быстром «прохождении» материала пропущенных уроков возникает проблема неотработанных моментов базовой части материала. Собственно говоря, страдает фундамент построения линии обучения.

2. Технические средства обучения

Технические средства представлены не только устройствами, но и разнообразными технологиями. Весь арсенал разнообразного оборудования, техники, устройства и,

безусловно, программного и информационного обеспечения прочно вошел в сопроводительную часть организации рабочих мест и учебного процесса на различных ступенях (как в школьном, так и в вузовском и средне-специальном).

2.1 Использование в образовании

Технические средства обучения повсеместно входят в работу преподавателей различных дисциплин. Компьютеры, планшеты, интерактивные доски и проекторы давно стали частью учебного процесса. Весь этот арсенал используется как эпизодически, так и регулярно. В зависимости от дисциплины выбор того или иного технического средства актуально как технология или как аппаратная часть. Технологические особенности сопровождения учебного процесса в зависимости от изучаемого предмета могут, как ни странно, весьма обширно пересекаться. Тем не менее, такой предмет, как информатика и ИКТ, привносит основополагающие моменты для доминирующей роли технических средств, которые эффективно задействуются при обучении.

Системы автоматического тестирования как информационно-коммуникационные технологии изначально проектировались для упрощения проведения олимпиад по программированию различного уровня. До их написания были ручные «проверки», приходилось в командной строке сравнивать выходные данные с «эталонными», которые выдавало авторское решение. Автоматизация тестирования задач олимпиад различного уровня значительно упростила и облегчила процесс проведения мероприятий соревновательного характера.

Система тестирования в НГУ начала разрабатываться с 1998 года. Работы по архитектуре системы [9] были сформированы в соответствии с требованиями, предъявляемыми к автоматизированным системам тестирования знаний для проведения олимпиад по программированию различного уровня [10, 11, 12].

Надо сказать, что применялась не только система тестирования НГУ, также использовались системы тестирования ЯндексКонтест и Codeforces [13]. ЯндексКонтест в качестве платформы для решения задач по программированию был нами опробован, в частности, для проведения отборочных мероприятий для Летней школы юных программистов. К недостаткам можно отнести не очень удобный сервис по созданию логинов и паролей. При этом надо отметить, что организаторы сервиса откликаются на просьбы и предложения по улучшению и активно идут навстречу пользователям.

Codeforces изначально создана для поддержки соревнований программированию. К достоинству данной системы можно отнести постоянно поддерживающийся и

развивающийся сервис. В частности, для создания новых соревнований очень удобен механизм Polygon.Codeforces.

У каждой из систем есть свои многочисленные достоинства и недостатки. Можно пренебрегать некоторыми неудобствами в работе с программами, можно пытаться дописывать к собственной системе функционал, который стал необходимостью.

Codeforces представляет собой платформу для соревновательного программирования,

которая объединяет как уже состоявшихся разработчиков, так и студентов и даже школьников с со всего мира.

Особенностью Codeforces являются регулярно проводимые соревнования, число которых составляет в среднем 6 в месяц. На основе результатов соревнований участники получают как отчёты по тестированию написанных алгоритмов, так и оценку их навыков, которая затем формирует общий рейтинг участника, который позволяет анализировать текущие результаты и свой прогресс. Регулярно проводимые соревнования дают возможность соревноваться с лучшими в спортивном программировании.

Также платформа представляет инструменты для обсуждения решений и обмена опытом пользователями, что сформировало активное сообщество программистов. На сайте

проекта можно ознакомиться с редакторскими и обучающими материалами.

Представленные задачи разнообразны, но всё же ориентированы на имеющих алгоритмическую базу пользователей, порог вхождения для полноценного использования платформы высокий.

С технической точки зрения, Codeforces поддерживает все популярные языки программирования, но в целом их набор достаточно ограничен. Отправленные решения задач проверяются достаточно оперативно. Однако, во время пиковых нагрузок возможны сбои в доступе к серверам.

Яндекс.Контест позиционирует себя как платформа для проведения соревнований и проверки решений задач по программированию. ЯКонтест подходит для начинающих, совмещая простой и интуитивно понятный интерфейс с гибкой настройкой проводимых соревнований. Однако, последние проводятся только по запросу и вся гибкость настроек, которые могут оценить участники, ложится на человека, подготавливающего туры соревнований в системе. Также плюсом является широкий выбор языков программирования для использования. С технической точки зрения, система надёжна и стабильно работает всё время, сбои бывают редко, отправленные решения в соревнованиях проверяются очень быстро. ЯКонтест ориентируется на обучение, что часто выливается в интеграцию с

образовательными программами – система часто используется в различных курсах и локальный соревнованиях.

Рекомендации по выбору следующие: Codeforces рекомендуется пользователям, имеющих в своём арсенале навыков основы алгоритмики, для тех, кто хочет получить больше опыта решения сложных задач различных соревнований и тех, кто хочет видеть свой прогресс в системе и любит повышать свой рейтинг как оценку соответствующих навыков. Яндекс.Контест подходит для начинающих программистов, студентов и школьников, тех, кто только начал проходить обучающие курсы, для тех, кто ищет стабильную платформу для практики. Также система хорошо подходит для организации локальных соревнований.

3. Использование системы в учебном процессе

Отсутствие ряда механизмов, которые заведомо не были предусмотрены в автоматизированной системе тестирования, не смогли стать препятствием для применения системы в регулярном учебном процессе в качестве дополнения к урочной системе. Система предназначена для проверки только решений в одном файле. Использование собственных библиотек и тестирование программ, состоящих из нескольких модулей в системе проверить нельзя. В автоматизированной системе проверки решений входные данные считываются как тесты из файла и выходные данные решения должны быть записаны в файлы. Начинающие программировать не всегда умеют грамотно работать с файлами данных. Поэтому предусмотрено автоматическое перенаправление ввода и вывода из стандартного потока в файл. Если не проверять решение (код программы), то может оказаться, что решение выдало правильный ответ, но задача решена не та, которая была поставлена.

Тем не менее, система оказалась удобной и результативной. Возможность проработки теории и закрепление материала решением подобранных задач в индивидуальном режиме, благодаря круглосуточной работе системы, позволяющей получить доступ к сдаче и проверке заданий в удобное время при самостоятельной работе.

Присутствует соревновательный эффект при отслеживании успехов соучеников на мониторинге сданных задач. Эффективность использования и ряд других положительных моментов весьма ощутим для отслеживания «обратной связи» с обучающимися, для индивидуального подхода к контролю знаний, а также для формирования банка задач по той или иной теме.

Решения задач проверяются на наборе тестов. Тесты, как правило, генерируются на основе эталонного решения задачи с учетом алгоритмических особенностей задачи. Первичная проверка на небольшом количестве тестов из примера в тексте задачи проверяет,

компилируется или нет программа. За эти тесты можно не начислять первичные баллы. Далее прогон решений задач возможно осуществлять на полном наборе тестов с учетом всех веток алгоритма. Этот подход значительно экономит время при первичной проверке заданий. Уделять больше внимания разбору алгоритмов, правилам оформления решения на языке программирования, проверке понимания текста программы – мечта каждого преподавателя. В качестве возможных языков программирования, проверка решений задач на которых возможна на сервере, который обрабатывает проверку правильности решения задач, как классические Паскаль и Си, C++, так и популярные на сегодняшний день Питон и Ява.

Проверка заданий может осуществляться удаленно. Возможна помощь ученикам во внеурочное время. Всегда можно написать письмо, сообщить о видимой ошибке в программе или выслать тест, на котором программа не проходит проверку в системе. Это зачастую помогает найти ошибку в программе самостоятельно, выяснить, какую веточку алгоритма не учел сдающий в своем решении или, возможно, неверно выбрал тип данных при реализации алгоритма. Система создавалась для проведения олимпиад. Как правило, в олимпиадах по программированию доступ к набору тестов имеет только жюри, участникам тесты не доступны. Сейчас практикуются задачи с «открытыми тестами». Но пока в настоящее время проблема неверного ответа на каком-либо тесте при проверке в системе тестирования решается только посредством прямой выдачи теста обучающемуся (в личной почтовой переписке или иным способом передачи файлов с данными).

Новый материал, который изучается на семинарах, отрабатывается решением задач на практике. При этом начинающие программировать нуждаются в непрерывном контроле со стороны преподавателя. Зачастую элементарное отсутствие навыков сохранять текст программы, которая успешно скомпилирована и выдала правильные результаты на прилагаемых примерах, не бывает сохранена. Момент привития навыков культуры программирования весьма непростой. Структурированность текста программы, введение «говорящих» переменных, выбор подходящих типов данных в конкретных реализациях задач – далеко не полный список того, на что необходимо сфокусировать преподавателю взгляд при проверке решений. Технологически сложно и затратно по временным категориям отслеживать все моменты во время занятия.

Анализ правильности решений как алгоритмических, так и технических в свободном режиме, во внеурочное время позволяет без цейтнота во время занятий составить список м присущих большинству решений ошибок реализации, которые надо доработать и пояснить всем, так и выявить индивидуальные моменты для каждого (например, не структурированно оформленное решение). Для фиксирования использования тех или иных структур данных

или алгоритмов создаются тематические туры. В наборе существует, например, тема «Задачи на циклы», также про алгоритмы и структуры данных есть набор по теме «Полный перебор». Данный прием позволяет сконцентрировать внимание не только на решение задачи, но и на алгоритмическую составляющую, методы и структуры данных, которые должны быть отработаны в текущий момент изучения программирования.

Помимо этого, предусмотрено создание специализированных туров, например, для проведения контрольных и проверочных работ. В системе можно создать 2-3 варианта заданий. Например, один вариант контрольной работы: сразу виден результат.

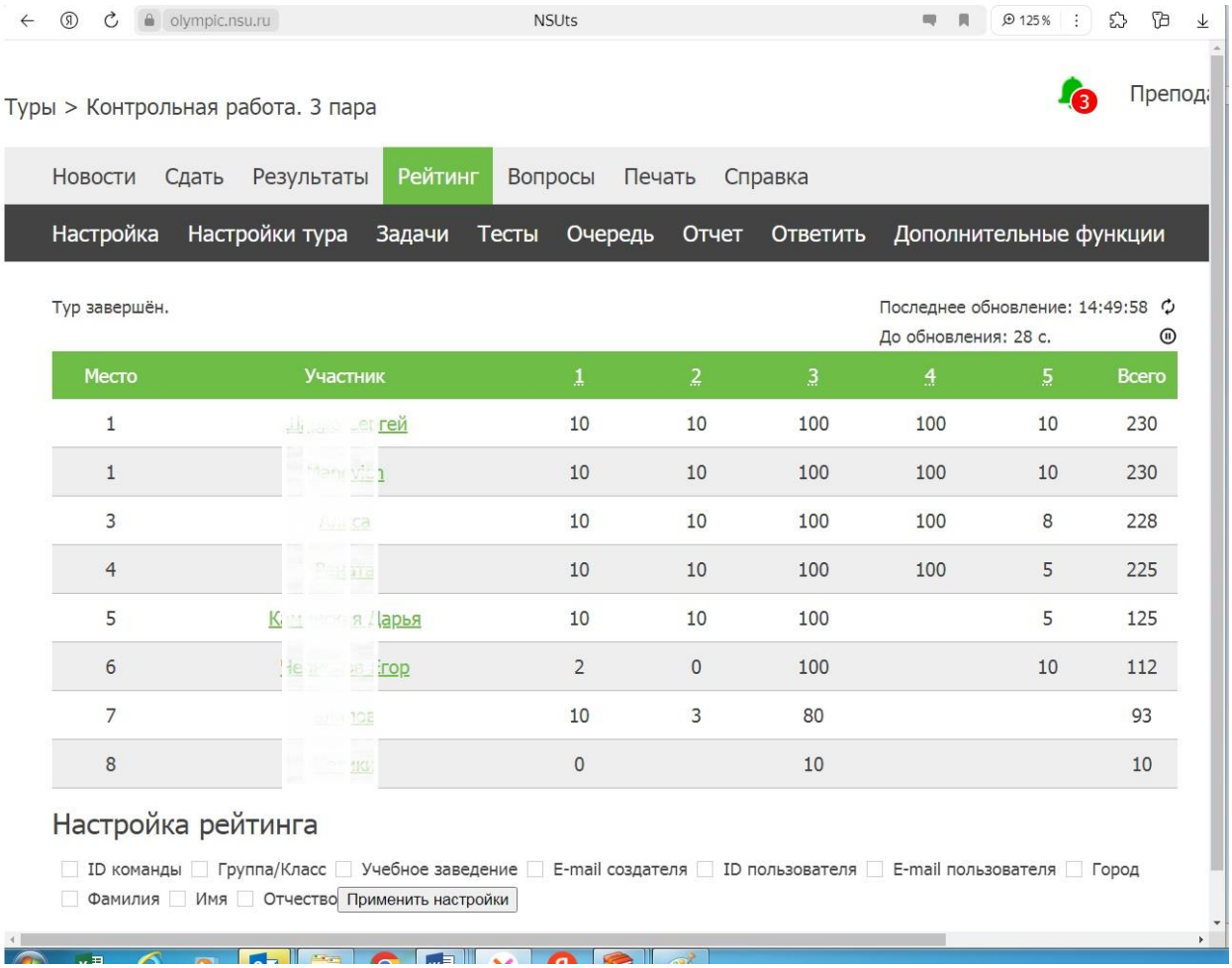


Рис. 2. Страница результатов выполнения регулярного задания всех участников

Далее у каждого из участников можно посмотреть его решения и тесты, которые не прошли. Отражается вся последовательность посылаемых в системе решений задач для тестирования. По данной страничке можно отследить недостатки хода решения, а уточнить, в чем заключается ошибка программы можно, посмотрев конкретный файл.

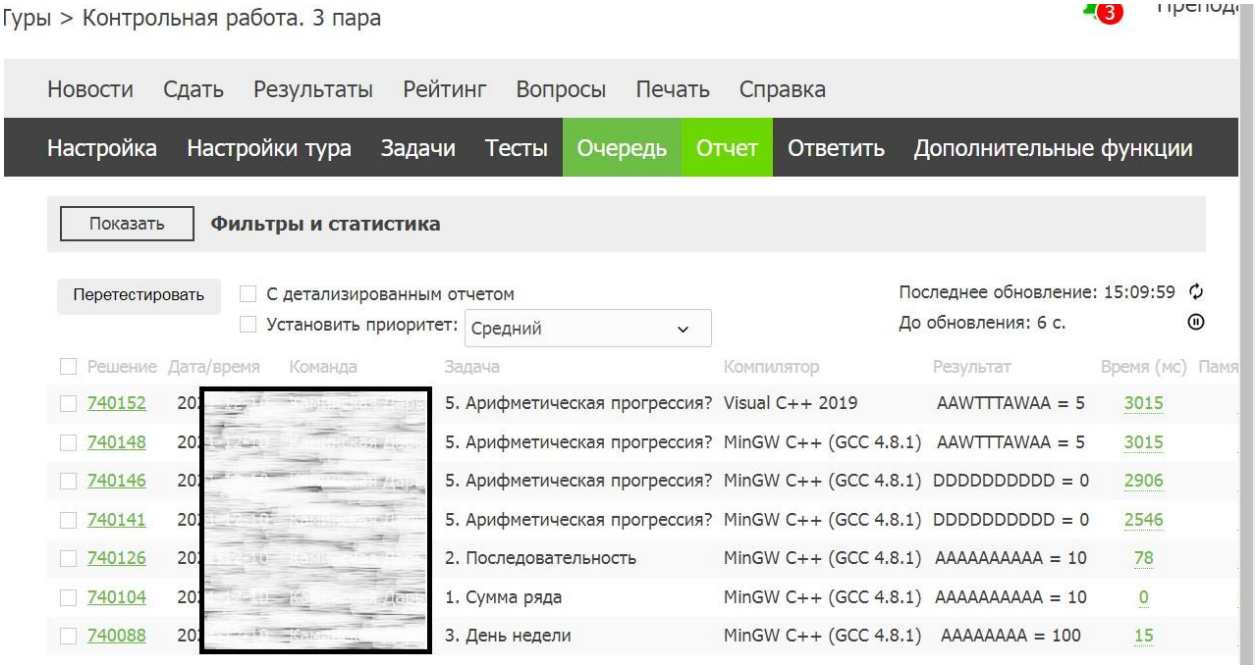


Рис. 3. Страница результатов выполнения контрольной работы конкретного участника

Весьма интересен опыт воспитательный: выявление списываний с использованием проверки на плагиат в системе совершенно по-иному заставляет работать взаимопомощь при отработке учебного материала. Система проверки заданий может быть настроена по двум принципам. Первый вариант, когда решение задачи принимается только при прохождении всех тестов. Второй вариант позволяет ввести балльную систему в зависимости от количества пройденных тестов. Можно поставить неограниченное количество сдач заданий в систему. Данный момент учит не бояться делать ошибки, когда решаешь задачу.

Помимо проверки правильности работы решений на предложенных тестах для продвинутых учащихся вводится ограничение по времени работы программ, что необходимо для анализа эффективности выбранных для решения задач алгоритмов. В качестве примера можно понять, что решение полным перебором какой-либо задачи, даже с использованием эвристик, не может быть эффективнее, чем применение метода динамического программирования, основанного на хранении в таблице промежуточных данных (классическая задача «Рюкзака»).

Практически в каждой тематической линии есть задания различного уровня (базовый, основной и продвинутый). Минимум, что надо выполнить для получения отметки – задания из базового набора. Базовые задачи подразумевают правильно выбранные структуры данных и типы данных при написании программы, отладки для контроля выдаваемых результатов, умение правильно прочитать текст задания. В некоторых случаях система тестирования

позволяет даже отключить обязательное умение работы с файлами. Осуществляется данная возможность очень просто: в настройках удобно отметить галочкой необходимость тестирования без осознанного владения начинающего программировать приемами работы с файлами. Для решения задач так называемых «основных» заданий требуется написать самостоятельно новые функции, найти ходы, формулы. Для решения набора задач «продвинутого» уровня обучающийся, как правило, осваивает дополнительный материал, анализирует существующие подходы, комбинирует и адаптирует к решению задач известные алгоритмы, подбирает структуры данных и т.д. Таким образом, от простого – к сложному, через осознание теоретического материала на практике, анализ знаний и поиск новых решений идет наработка компетенций.

И этот список – далеко не полный перечень преимуществ, который был выявлен при внедрении автоматизированной системы проверки в учебный процесс.

4. Заключение

Технические средства в качестве дополнения к базовым методикам на уроках информатики и ИКТ используется в течение нескольких лет на занятиях не только со студентами, но и со школьниками. Одним из них являются системы автоматического тестирования. Данный ресурс является эффективным дополнением к регулярным урокам. При анализе результативности применения системы тестирования можно отметить несомненное улучшение качества усваивания учебного материала теми группами обучающихся, в которых данная методика практиковалась. Высокий результат всех групп по ЕГЭ, согласно опросам обучающихся, сложился благодаря тому, что они изучали программирование и приемы тестирования, отрабатывали на практике решение задач различного уровня сложности с использованием автоматизированной проверки решений с последующим анализом программ в индивидуальном порядке. Таким образом, решаются дидактические задачи посредством получения навыков решения, закрепления пройденного материала на практике и проверки индивидуальных решений в системе тестирования как при классно-урочной системе, так и при самостоятельной и коллективной работе. Поиск самостоятельного решения благодаря широкому набору задач по каждой теме от типовых до сложных эффективно способствует формированию исследовательского подхода в обучении. Организация учебного процесса через практическую деятельность по анализу и решению разноуровневых задач в индивидуальном режиме способствует мотивации и вовлечению в изучение дополнительных глав предмета. Регулярный контроль знаний и осваивание самостоятельной работы помогают более комфортно осуществлять обратную связь и

анализировать рефлексивные навыки обучающихся. При таком подходе наблюдается явная корреляция с иерархией Блюма. Наряду с другими методическими приемами, использование технических средств обучения, в частности, системы для автоматической проверки решений задач, способствует умению планировать свое время при работе над домашними заданиями. Стрессоустойчивость при самостоятельной работе над реализацией работающей программы формируется непосредственно при умении находить ошибки в своих решениях. Кроме технических навыков формируется неформальный, а личностный подход к обучающемуся. Оценка в системе тестирования – это еще и обратная связь от преподавателя, который подталкивает к нахождению решения задач в процессе отладки, подсказывает, на какой момент надо обратить внимание (алгоритм или техническое решение, или выбор структуры данных). Самоорганизация также имеет не последнее значение. Методика построения обучения с технологическими включениями выстраивается на основе подтвержденных показателей как результативности обучения, так и обратной связи от обучающихся.

Таким образом, использование систем тестирования в качестве технического средства, дополняющего образовательный процесс, зарекомендовало себя как эффективный инструмент. Он позволяет комплексно отслеживать усваивание учебного материала, гибко настраивать содержание проверочного материала, оптимизирует трудозатраты при подготовке и проверке заданий.

Список литературы

1. Bloom B.S. Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. N.Y., – 1956, – s.111.
2. Интернет-ресурс https://medium-com.translate.goog/@niall.mcnulty/using-blooms-taxonomy-in-lesson-planning-9e0dd1fd548f?_x_tr_sl=en&_x_tr_tl=ru&_x_tr_hl=ru&_x_tr_pto=rq#:~:text=How%20to%20incorporate%20Bloom's%20Taxonomy,analysing%2C%20evaluating%2C%20or%20creating. Ниалл МакНалти. Использование таксономии Блума при планировании уроков. Дата обращения 15.01.2024 г.
3. Марчук, А. Г. Новосибирская школа юных программистов / А. Г. Марчук, Т. И. Тихонова, Л. В. Городняя // Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы. SORUCOM.2006 : Материалы международной конференции, Петрозаводск, 03–07 июля 2006 года / Ответственный редактор: Н.С.Рузанова. Том 2. – Петрозаводск: Петрозаводский государственный университет, 2006. – С. 117-124. – EDN STOMTH.
4. Насибулов, И. А. Роль метода проектов в изучении информатики для формирования навыков исследовательской деятельности / И. А. Насибулов, Т. И. Тихонова // Профильное образование

и специализированное обучение: перспективы развития в цифровом пространстве : Сборник материалов Всероссийской научно-методической конференции с международным участием, Новосибирск, 10–11 декабря 2022 года / Отв. редактор В.В. Петров. – Новосибирск: Новосибирский национальный исследовательский государственный университет, 2022. – С. 52-56. – EDN NUGBIY.

5. Марчук, А. Г. Традиции в системе подготовки творческой молодежи / А. Г. Марчук, Т. И. Тихонова // Компьютерные инструменты в школе. – 2008. – № 2. – С. 3-12. – EDN TBGMVF.
6. Тихонова, Т. И. Новосибирская летняя школа юных программистов им. А.П. Ершова (21 июля - 3 августа 2014 г., Новосибирский Академгородок - "Белый камень") / Т. И. Тихонова // Системная информатика. – 2015. – № 5. – С. 75-104. – DOI 10.31144/si.2307-6410.2015.n5.p75-104. – EDN VSZWAD.
7. Марчук, А. Г. Летняя школа юных программистов - этап становления школьной информатики / А. Г. Марчук, Т. И. Тихонова // От информатики в школе к техносфере образования : Сборник научных трудов Международной научно-практической конференции, Москва, 09–10 декабря 2015 года / Российская академия образования, Московский городской педагогический университет, Московский педагогический государственный университет. – Москва: ООО "Издательство "Научная книга", 2016. – С. 243-249. – EDN VHZFWJ.
8. Тихонова, Т. И. Научно-исследовательская деятельность как метод обучения и социализации / Т. И. Тихонова // Профильное образование и специализированное обучение: стратегия, тактика и технология развития в поликультурном пространстве : Сборник материалов Всероссийской научно-методической конференции с международным участием, Новосибирск, 09–10 декабря 2023 года. – Новосибирск: Издательско-полиграфический центр НГУ, 2023. – С. 58-62. – EDN RBWMXF.
9. Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г. Разработка систем автоматизированной оценки заданий по программированию // Журнал «Системная информатика». 2017. № 11. С. 91–116.
10. Боженкова Е. Н., Иртегов Д. В., Киров А. В., Нестеренко Т. В., Чурина Т. Г. Автоматизированная система тестирования NSUts: требования и разработка прототипа // Вестн. НГУ. Серия : Информ. технологии. 2010. Т. 8, Вып. № 4. С. 46–53.
11. Чурина Т. Г., Иртегов Д. В. Требования к автоматической системе тестирования знаний : труды VI Междунар. конф. «Интеллектуальные технологии в образовании, экономике и управлении». Воронеж, 2009. С. 309–317.
12. Боженкова Е. Н., Воронков А. Д., Иртегов Д. В., Конышева Е. Н., Черненок С. А., Чурина Т. Г. Модель разграничения прав доступа в системе автоматизированной проверки корректности программных приложений // Вестн. НГУ. Серия : Информ. технологии. 2011. Т. 9, Вып. № 4. С. 79–85.
13. https://habr.com/ru/companies/spbifmo/articles/538918/. Дата обращения 15.08.2025 г.

14. Воронов М.П., Часовских В.П. Методика оценки эффективности электронных образовательных систем // Современные наукоемкие технологии. 2022. № 10 (часть 1) С. 120-124 URL: https://top-technologies.ru/ru/article/view?id=39357 (дата обращения: 21.08.2025). DOI: https://doi.org/10.17513/snt.39357