Formal methods and tools for evaluating cryptographic systems security

Alexandra A. Savelieva

Scientific Advisor: Prof. Sergey M. Avdoshin

Cryptographic system operates in an environment that imposes restrictions on the types of attacks that the system is exposed to. Classically, the research has mostly focused on information system security as a whole, whereas cryptographic tools evaluation techniques have not received as much attention. The main thread of our work is the development of formal techniques to analyze the security of cryptographic systems based on varying attacks. The second main thread is the development of software tools to facilitate the process of cryptosystem efficiency assessment by computer security specialists.

***Index Terms*— cryptographic system, cryptanalysis, classification of attacks, discrete logarithm problem.**

I. Introduction

Bruce Schneier, a well-known cryptographer, declares in [1] that the term "security" does not have meaning unless you can answer such questions as "Secure from whom?" or "Secure for how long?". This statement applies to security systems in general as well as to their essential component – cryptographic systems. Classically, the research has mostly focused on information system security as a whole, whereas cryptographic tools evaluation techniques have not received as much attention. Our paper aims at providing a formal, methodical way of analyzing the cryptographic systems security.

The process of cryptosystem efficiency assessment can be described using the scheme in Fig. 1. Each step is directed at answering a specific question:

- Step 1: What cryptosystem is the object of attack?
- Step 2: Who wants to attack the cryptosystem?
- Step 3: Which attack techniques are most likely to be used to break the cryptosystem?
- Step 4: Is the cryptosystem capable of withstanding such attacks?
- Step 5: Does the cryptosystem provide sufficient security in the given context?

Steps 1 to 3 imply modeling threats to a cryptographic system in a given context. The environment typically imposes restrictions on the attack scenarios that the cryptographic systems are exposed to. A simple model of a code-breaking scenario is introduced in Fig. 2. It includes three elements, each of them contributing to overall picture of a threat. We propose three classification schemes: of attackers, of cryptosystems, and of attacks. With these multicriterion classification schemes, we can build a formal model of the cryptosystem that we are investigating (Step 1 in Fig.1), as well as models of the attackers who can potentially threaten the system (Step 2 in Fig. 1). There is a dependency between model parameters of a cryptosystem and possible types of attacks that can be applied to it. Similarly, levels of skill, access, risk aversion, money, etc. define the attacks that an adversary can undertake [1]. Having such formal representations of the system we are investigating and of the potential intruders, we can proceed to Step 3 in Fig. 1 and determine the set of attacks that the cryptosystem is exposed to, as well as their probability.

The next step (Step 4 in Fig.1) is analyzing the cryptographic system resistance to the specific types of attacks defined at Step 3. The purpose of our work is to provide a specialist with convenient tools to perform this analysis. We developed a software system to facilitate the process of asymmetric cryptosystems evaluation by means of modern cryptanalysis techniques. We describe some results achieved using the tools to improve the algorithms for discrete logarithm computation.

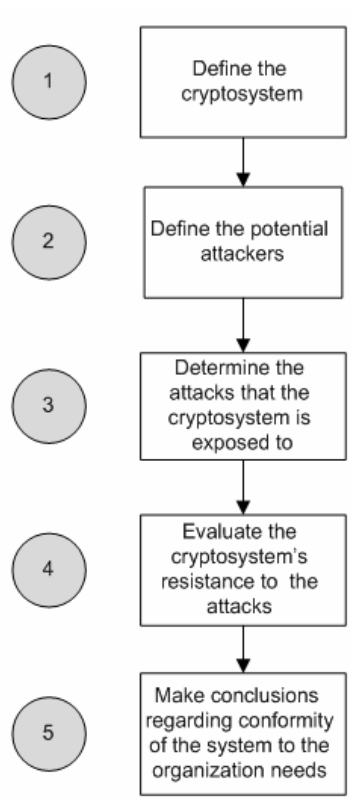

Fig. 1.Cryptosystem efficiency assessment process

Finally, Step 5 in Fig.1 involves using various risk analysis

techniques to evaluate the data obtained during Steps 1-4. The results can be integrated into the overall system security assessment report, thereby increasing the accuracy of risk analysis.

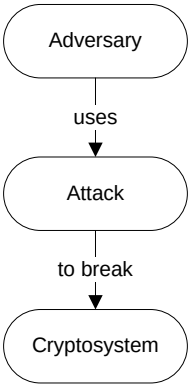

Fig. 2. Code breaking scenario

II. MODELING THREATS

Each cryptosystem has a set of attacks that is applicable to it and a set of attacks that is not. We can also assume that the adversary is most likely to choose the attack with the maximum benefit for a given cost, or choose the least costly attack that gives them a particular benefit [2]. Thus, a formal model of the cryptosystem coupled with formal models of the adversaries will yield a set of attacks that the cryptosystem is exposed to. The next subsections describe the classifications we suggest as a basis for modeling the components of a code-breaking scenario in Fig. 2.

A. Classification of cryptosystems

Examining a cryptosystem and trying to imagine all the possible threats against it is critical to estimating its security. We need to judge a risk based not only on who is likely to attack the system and what they want, but on exactly what cryptographic tools are being attacked.

There are various classification schemes of cryptosystems available in modern literature. For example, Ueli Maurer's idea is to distinguish cryptosystems by the number of keys used for data processing, i.e. unkeyed, single-keyed, and double-keyed cryptosystems [3]. Gilles Brassard's scheme [4] has to do with the secrecy of algorithm. Yet another classification is suggested by Friedrich L. Bauer [5]. However, neither of these classifications reflects all the properties necessary to identify a cryptosystem in practice. We propose a multicriterion classification scheme that includes the criteria mentioned above along with several new criteria. The set of criteria is useful for building a parametric model of a cryptosystem and for determining the set of attacks applicable to it. We suggest that the cryptographic tools should be identified using the following criteria:

- By secrecy of the algorithm
  - Restricted
  - General
- By the number of keys
  - Unkeyed, including hash-functions and pseudorandom generators
  - Single-keyed, or symmetric
  - Double-keyed, or asymmetric
  - Multiple-keyed, or threshold scheme for secret sharing
- By breakability
  - Theoretically unbreakable
  - Provably unbreakable
  - Supposedly unbreakable
- By the means of implementation
  - Software
  - Hardware
  - Software and hardware
- By certification
  - Certified
  - Uncertified

B. Classification of attackers

Types of attackers that you are defending against define the sensible type of security. Predicting most likely attackers and understanding them gives a clue to how they might attack the assets protected by the cryptosystem [2]. Bruce Schneier suggests using motivation as a key parameter to identifying an adversary; this results in the following classification scheme:

- opportunists:
- emotional attackers
- friends and relatives
- industrial competitors
- the press
- lawful governments,
- the police
- national intelligence organizations

There is an *m : n* relationship between the types of attackers and the types of attacks, i.e. a single attacker can undertake many different attacks, and a single type of attack can usually be launched by a number of different attackers. Being excellent for high-level analysis, Schneier's classification however provides no clear mapping between the type of attacker and the attacks they can use. We designed a new scheme to give a more precise definition to the types of adversaries in terms of different model parameters. We suggest that the attackers should be identified using the following criteria:

- By equipment
  - PC
  - Network
  - Supercomputer
- By final objective
  - Discovering a vulnerability
  - Total break
- By access
  - Insider
  - Outsider
- By expertise
  - PC user
  - Mathematician

    - Software developer
    - Physicist/electrical engineer
    - Psychologist aware of social engineering techniques
- By initial knowledge on the cryptosystem
    - User of the cryptosystem
    - Designer of the cryptosystem
- By manpower
    - Individual
    - Team

C. *Classification of attacks*

There are a lot of computer system attack classifications and taxonomies, e.g. those suggested in [6], [7], [8], [9]. However, they are designed to describe intrusions into a computer system, whereas the object of our research is a specific type of attack – code-breaking.

The fundamental classification of attacks by access to plaintext and ciphertext introduced by Kerckhoffs [10] is no longer complete since it does not include a new powerful cryptanalysis technique called Side-Channel attacks [11]. Lars Knudsen [12] classifies the attacks based on the result. We suggest a set of criteria that allows to identify an attack and relate it to the types of adversaries who can potentially use and the cryptosystems it is applicable to. We categorize the attacks as follows:

- By access to plaintext and ciphertext
    - Ciphertext-only
    - Known-plaintext
    - Chosen-plaintext
    - Adaptive-chosen-plaintext
    - Side-channel
- By control over the enciphering/deciphering process
    - Passive
    - Active
- By the outcome
    - Total break
    - Global deduction
    - Instance (local) deduction
    - Information deduction
    - Distinguishing algorithm
- By critical amount of resources
    - Memory
    - Time
    - Data
- By applicability to various ciphers
    - Multi-purpose
    - For a certain type of ciphers
    - For a certain cipher
- By tools and techniques
    - Mathematics
    - Special-purpose devices taking physical measurements during computations
    - Evolution programming techniques
    - Quantum computers
- By consequences
    - Breach in confidentiality
    - Breach in integrity
    - Breach in accessibility
- By parallelizing feasibility
    - Distributed
    - Non-distributed

The rationale for the criteria choice is described in a survey of modern techniques of cryptanalysis [13].

III. SOFTWARE TOOLS

The software tools CRYPTO [14] are designed as a means for conducting research in information security, number theory, and algebra. With the advent of new attack techniques and computer power growth, cryptographic algorithms security is constantly reducing. To decrease the possible damage, it is necessary to regularly check the cryptoalgorithms security. This includes both development of new cryptanalytical methods and improving the efficiency of existing methods.

CRYPTO consists of two components: a dynamic-link library DESIGNER, and an application ANALYST (see Fig. 3). ANALYST provides a friendly graphical user interface to access functions of DESIGNER. DESIGNER is a high-performance, portable C++ library providing the necessary elements to design and evaluate modern techniques for cryptanalysis of ciphers based on factorization and discrete logarithm problems. The implementation makes use of NTL (a

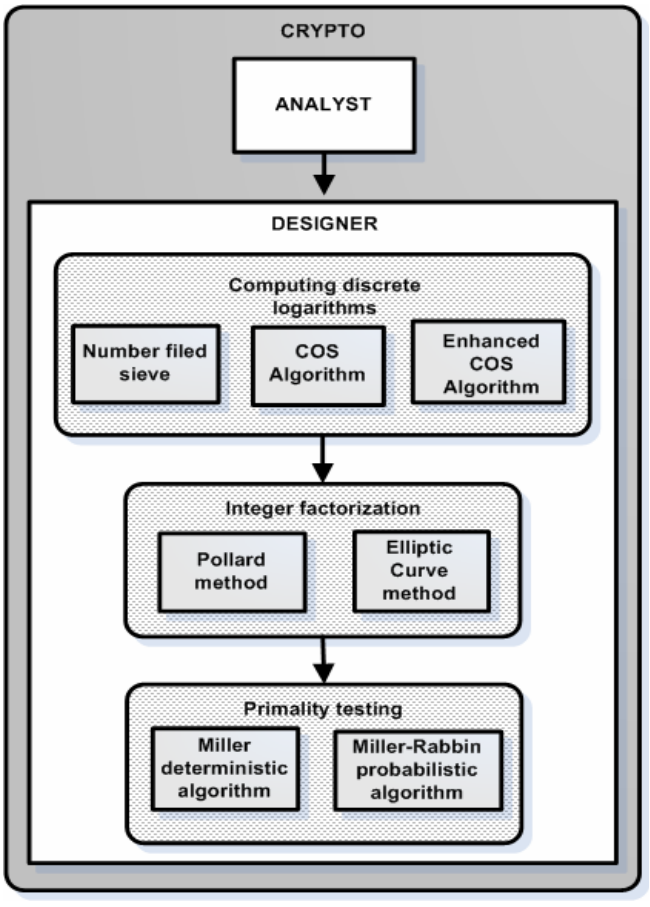

Fig. 3. CRYPTO structure

Library for doing Number Theory) written and maintained by Viktor Shoup [15].The rationale for the core library is its functionality, performance, and portability.

CRYPTO implements the most effective modern algorithms for factorization and discrete logarithm computation which have subexponential time complexity. Factorization algorithms include Pollard algorithm [16] and "ECM" (elliptic curve factorization method by Lenstra) [17]. Before factorization, it is necessary to check if the integer is composite. This is achieved through using one of primality tests: Miller-Rabin's probabilistic algorithm [18] or Miller's deterministic algorithm [19].

Discrete logarithm computation techniques include Number Field Sieve [20], COS (Coppersmith-Odlyzhko-Shroepel algorithm) [21] and an improved version of COS. Enhanced COS uses an efficient algorithm that we designed for solving linear systems over residue rings [22], [23]. Its complexity is equivalent to that of the Gauss method [24] for solving linear systems over finite fields. Table 1 illustrates the time complexity of our algorithm compared to other methods when solving a system of n linear equations with m unknowns in $\mathbf{Z}_p$ (where $p = \prod_{k=1}^{t} q_k^{\alpha_k}$). The new method significantly reduces the complexity of algorithms for discrete logarithm computation. A detailed description of the algorithm is in [22].

TABLE I
ALGORITHMS FOR SOLVING LINEAR SYSTEMS OVER RESIDUE RINGS

Algorithm	Complexity
Our method	$O\left(n \cdot (nm + \log p)\right)$
Reduction to prime fields	$O\left(n \cdot (n \cdot m \cdot \sum_{k=1}^{t} \alpha_k + \log p) + \sqrt{\ln p \ln\ln p} \cdot e^{\sqrt{\ln p \ln\ln p}}\right)$
Transformation to Diophantine equations	$O(n^2 m^2 \log p)$

IV. CONCLUSION

The proposed classifications provide a formal methodology for analyzing the security of cryptosystems. Model-based analysis is a part of the five-step process designed to focus on the specific aspects of cryptographic systems security. We expect that the proposed approach will be of value to security professionals, application developers, software vendors or anyone else with an interest in information security.

One direction of our future work is the development software tools to automate the steps 1-3 in Fig.1. This involves formulating mathematical rules to define the dependency between the parameters of a cryptosystem model and the applicable attacks on the one hand, and the parameters of an attacker model and the types of attacks that they are likely to use, on the other hand.

Another direction of our work is the using of 'CRYPTO' software tools to design new algorithms and improve present methods for factorization and computing discrete logarithms. We are also working on extending the library to include modern techniques to analyze the security hash-functions as well as asymmetric cryptosystems.

REFERENCES

[1] Schneier B. Modeling security threats // Dr. Dobb's Journal, December, 1999.

[2] Schneier B. Beyond Fear. Thinking Sensibly about Security in an Uncertain World. Copernicus Books (September 2003)

[3] Oppliger R. Contemporary Cryptography. Artech House Publishers , 2005, 510 p.

[4] Brassard J. Modern Cryptology. Springer-Verlag, Berlin - Heidelberg, 1988. - 107 p.

[5] Decrypted Secrets: Methods and Maxims of Cryptology: FL Bauer: Springer-Verlag Telos; 2nd Rev&Ex edition (February 2000), 470 p.

[6] Landwehr C. E., Bull A. R. A taxonomy of computer program security flaws, with examples // ACM Computing Surveys, 26(3): p. 211–254, September 1994.

[7] Lindqvist U., Jonsson E. How to systematically classify computer security intrusions. // IEEE Symposium on Security and Privacy, p. 154–163, Los Alamitos, CA, 1997.

[8] Weber D. J. A taxonomy of computer intrusions. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1998.

[9] Paulauskas N., Garsva E.. Computer System Attack Classification // Electronics and Electrical Engineering 2006. nr. 2(66)

[10] Kerckhoffs A. La cryptographie militaire // Journal des sciences militaires, vol. IX. P. 5-38, Jan. 1883, (P. 161-191, Feb. 1883).

[11] Zhou Y., Feng D. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. // Available at: http://eprint.iacr.org/2005/388.pdf

[12] Knudsen L. R., Mathiassen J. E. A chosen-plaintext linear attack on DES // In Proceedings of Fast Software Encryption - FSE'2000, (Schneier B. ed.), Lect. Notes in Comp. Sci. Sprinler-Verlag, 2001. V. 1978. P. 262-272.

[13] Avdoshin S.M., Savelieva A.A. Cryptanalysis: current state and future trends// Information technologies. Moscow, 'Novye technologii', in Appendix to № 3, 2007, 35 p. (in Russian).

[14] Avdoshin S.M., Savelieva A.A. Tools for asymmetric ciphers analysis: Industrial registration certificate No. 10193 dated 18.03.2008 (in Russian).

[15] Library for doing Number Theory. Available at: http://www.shoup.net/ntl/ 06.02.2007

[16] Pollard J.M. A Monte Carlo method for factorization // BIT Numerical Mathematics 15(3), 1975, pp. 331-334.

[17] Lenstra Jr. H. W. Factoring integers with elliptic curves // Annals of Mathematics (2) 126 (1987), 649-673.

[18] Rabin M.O. Probabilistic algorithm for testing primality // Journal of Number Theory 12 (1980), no. 1, pp. 128–138.

[19] Miller G.L. Riemann's Hypothesis and Tests for Primality // Journal of Computer and System Sciences 13 (1976), no. 3, pp. 300–317.

[20] Gordon M.D. Discrete logarithms in GF(p) using number field sieve //SIAM Journal on Discrete Mathematics 6, no.1, 1993, pp/ 124-138.

[21] Coppersmith D., Odlyzko A., Schroeppel R. Discrete logarithms in GF(p) // Algorithmica. 1986. V. 1. - P. 1—15.

[22] Avdoshin S.M., Savelieva A.A. Algorithm for solving linear systems over residue rings // Information technologies. Moscow, Novye technologii'', 2006. № 2.- p.50-54 (in Russian).

[23] Avdoshin S.M., Savelieva A.A. Program for solving linear systems over residue rings. Certificate of official registration in the register of software No.2005612258 dated 02.09.2005, Federal Service For Intellectual Property, Patents And Trademarks. (in Russian).

[24] Waerden B. L. Algebra. Vol. 1, Springer-Verlag, Berlin, 1991.

Manuscript received April 1, 2008.

Alexandra A. Savelieva is a second-year MSc student of Software Engineering Department, State University – Higher School of Economics (e-mail: alexandra.savelieva@gmail.com; optional phone: +7 (962) -902-4310).

Research supervised by Prof. Sergey M. Avdoshin, Head of Software Engineering Department, State University – Higher School of Economics, Russia (e-mail: savdoshin@hse.ru; optional phone: +7(495)771-32-38).