Towards Computational Subsystem for Experimental Analysis of Security Protocols

A.S. Mikhailov
Department of Cybernetics
Moscow Engineering Physics Institute (MEPHI)
Moscow, Russia
e-mail: almikh@mail.ru

Abstract[1]

This work is about analysis of security protocol failures dependent on mathematical properties of computational and cryptographic algorithms. Detailed protocol models with inclusion of RSA and KMOV algorithms are compared with the idealized formal protocol model of the same security protocol. Idealized models are unsuitable for analysis of mathematical protocol failures. Detailed models are useful for experimental analysis of security protocols. The computational and cryptographic algorithms can be performed as sequences of elementary algebraic operations and simple auxiliary algorithms. A set of such operations and algorithms implemented and described as a special computational subsystem.

1. Introduction

Information exchange protocols over open computer networks are composed of messages and computational algorithms sequences. Cryptographic algorithms are used to ensure protocol security properties such as confidentiality, authentication, integrity, anonymity and e.t.c. In such cases we mean *cryptographic protocols* or *security protocols*. The unified conditions of protocols performance over the hostile environment named Dolev-Yo threat model were presented in paper [3]. Intruder has full control over environment. It means that intruder can intercept, substitute and block any messages. Intruder can be one of the protocol participants or he can control one or more participants. However intruder can't access private areas of environment. Strong secrecy of cryptographic algorithms is also assumed. Intruder can't break algorithms, for example it is impossible to find private cryptographic key of known public key. The Dolev-Yo threat model is still actual. The Internet global network is a good practical example of such environment.

In case intruder can successfully attack protocol without breaking cryptographic algorithms, we mean *protocol failures* [8]. In fact algorithms perform in distributed mode. Failures can depend on mathematical properties of computational algorithms operations.

The problem of protocol security proof is very important. The mainstream of researcher's efforts in this area is concentrated on developing formal methods. A number of different methods can be used for formal protocol modeling, such as state machines, logics, algebraic calculus and others [6]. A common trait of formal methods is idealization of protocols for modeling. Computational and cryptographic algorithms are frequently modeled as "black boxes". In these cases algorithms are presented in symbolic forms of their input and output values only. The inner structure of algorithm isn't taken into account at all. Thus it's theoretically impossible to find the protocol failures that depend on mathematical properties of algorithms.

A graphic example of protocol models with algorithm details compared with idealized formal model is presented in this work. Requirements to computational software for experimental performing of computation and cryptographic algorithms during protocol analysis are mentioned. Experimental algorithms performing can supplement the formal methods to overcome their restrictions for security protocols analysis.

2. Protocol Models

Let us view a simple security protocol for cyclic transferring of a confidential message between two participants A and B. User A creates confidential message m and encrypts it using public key of user B. Then user A sends the calculated ciphertext C_{ab} to user B. User B receives C_{ab} and decrypts it using his own private key. Then user B encrypts the plaintext m once again using public key of initiator A and sends the ciphertext C_{ba} to

Proceedings of the 8th International Workshop on Computer Science and Information Technologies CSIT'2006
Karlsruhe, Germany, 2006

user A. User A decrypts C_{ba} using his own private key and extracts the original confidential plaintext m.

The detailed protocol models with RSA [9] and KMOV [5] cryptographic algorithms are presented below. Then they are compared with an idealized formal protocol model. For presentation goals the visualization of protocol models is used. Adapted UML sequence diagrams can be used for protocol visualization. This approach provides intuitive protocols perception. Performed computational operations and communication events are represented along vertical line, called "life line" for each participant.

2.1. Using RSA Algorithm with Common Module

Let us assume that RSA algorithm is used to ensure confidentiality of message m. The visual protocol model with inclusion of computational operations of RSA algorithm is presented in Figure 1.

Both participants A and B use the same common module n. Public key PK_a of user A consists of values e_a и n. Public key of user B consists of e_b и n analogously. The d_a and d_b values are private user's keys SK_a and SK_b correspondently. Intruder can intercept values of C_{ab} and C_{ba} ciphertexts. Let us also assume that intruder knows the values of public keys e_a, e_b and common module n.

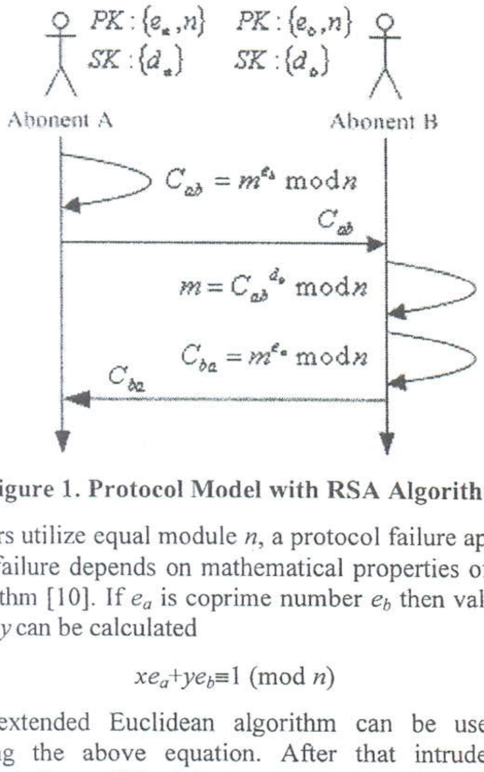

Figure 1. Protocol Model with RSA Algorithm

If users utilize equal module n, a protocol failure appears. This failure depends on mathematical properties of RSA algorithm [10]. If e_a is coprime number e_b then values of x and y can be calculated

$$xe_a+ye_b\equiv 1 \pmod n$$

The extended Euclidean algorithm can be used for solving the above equation. After that intruder can calculate the confidential message m

$$C_{ab}^{y}C_{ba}^{x} = (m^{e_b})^{y}(m^{e_a})^{x} = m^{(ye_b+xe_a)} = m \pmod n$$

2.2. Using KMOV Algorithm

The KMOV algorithm was named so after the first letters of the authors' second names [5]. It is a public key cryptographic algorithm based on elliptic curves over the ring Z_n and $n=pq$. The concept of KMOV algorithms is similar to RSA algorithm. Therefore KMOV is sometimes positioned as RSA analogue for elliptic curves. The protocol model with computational operations of KMOV algorithm is presented in Figure 2.

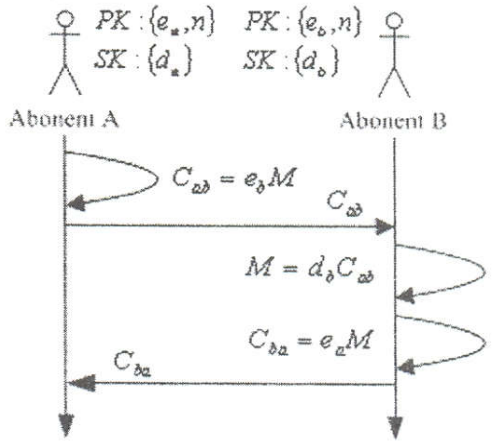

Figure 2. Protocol Model with KMOV Algorithm

An elliptic curve variant of protocol also has the analogous failure. Intruder can calculate point M that corresponds to confidential message m

$$yC_{ab}+xC_{ba} = y(e_bM)+x(e_aM) = (ye_b+xe_a)\,M = M$$

2.3. Idealized Protocol Model

Protocol idealization is required for analysis with formal methods. In most cases computational and cryptographic algorithms are presented as "black boxes". In this case algorithms are modeled in symbolic forms of input and output values only. The cryptographic key used can also be specified. The core of computational operations is out of scope at all. The sequences of computational operations are not taken into account. Moreover the type of mathematical apparatus isn't important too. The example of such idealized protocol model is presented in Figure 3.

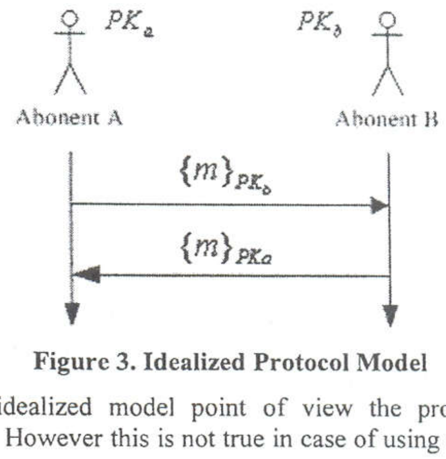

Figure 3. Idealized Protocol Model

From idealized model point of view the protocol is secure. However this is not true in case of using common module. Thus it's even theoretically impossible to find protocol failures dependent on mathematical properties of algorithms.

The problem of protocols modeling and analysis with inclusion of mathematical properties of used algorithms is a subject of hot discussions. However this problem is still open. Particularly one of modern investigation lines is concentrated on union of formal methods and provable security of cryptographic algorithms [1, 2, 6].

3. Experimental Analysis

From a practical point of view, computational and cryptographic algorithms must be taken into account for protocol analysis. The approach to using the distributed information system for experimental protocol analysis was presented in work [7]. Protocols run in special research environment created using the distributed information system based on network infrastructure. Protocols are modeled as computational processes synchronized by messages. Human involving is required for performing computational operations. Some users play legal roles of protocol participants. At the same time other users play intruder's role. Intruders can model attacks against of analyzed protocols. It means that intruders can receive copies of messages like in case of passive attacks. Moreover the system can stop running protocol and wait until the intrusion of an intruder's message as in case of active attacks. Legal protocol participants perform all required computational operations definitely. Intruders also have a possibility to do any computational operations in practice. This functionality is provided with the help of a special computational subsystem.

3.1. Computational Subsystem

Computational subsystem is meant for experimental performing of computational algorithms, including cryptographic algorithms. An important prerequisite is that shared key algorithms and hash-functions can be modeled as "black boxes". In fact criticism of abstract modeling is about public key algorithms in general. Public key algorithms can be presented as sequences of elementary algebraic operations and auxiliary algorithms. It is possible to define a restricted set of such operations and algorithms. The implementation of elementary operations interface and auxiliary algorithms make experimental performing of any computational and cryptographic algorithms possible. It's important to have a general set of elementary operations and auxiliary algorithms to avoid software modification for analysis of a new protocol.

The elementary operations for performing traditional (non elliptic curves) public keys algorithms are presented in Table 1.

The elementary operations with elliptic curves points are presented in Table 2.

Table 1. The Elementary Operations

Description	Operation
Addition	$(a+b) \bmod n$
Subtraction	$(a-b) \bmod n$
Multiplication	$(ab) \bmod n$
Division	$(a/b) \bmod n$
Exponentiation	$(a^b) \bmod n$

Table 2. The Elementary Operations of EC Points

Desciprion	Operation
Addition of two points	$(P+Q)$
Subtraction of two points	$(P-Q)$
Multiplication of point by integer	(kP)

It's good to clearly implement the interface of all these operations for usability goals. Auxiliary algorithms are required to perform more compounded computational and cryptographic algorithms. A set of auxiliary algorithms is presented below:

- String – Number: transformation and recovery
- String - EC point: transformation and recovery
- Random number generation
- Multiplicative inverse element
- Extended Euclidian algorithm
- Elliptic curve generation
- Order of elliptic curve
- Order of point of elliptic curve
- Hash-function
- Factorization
- Reduction by modulo

Also partial automation of computational and cryptographic algorithms is useful for experimental analysis goals. For example, the generation of RSA parameters can be automated given initial prime p and q.

3.2. Implementation

The size of cryptographic keys is unimportant for analysis goals. We assume that strong secrecy cryptographic algorithms are used in practice. For example it means that intruder can't break cryptographic algorithms with brute force attacks. Protocol failures depend on mathematical properties of algorithms for any size of keys and numbers. It is impossible to fix protocol failure by increasing the size of used keys. From a practical point of view it allows to use embedded numerical types of programming language for the implementation. So it makes no sense to support the real big numbers for analysis goals. As it was mentioned has-functions can be experimentally modeled as "black boxes". From a practical point of view they can be implemented as reductive algorithms (check-sum, for example). Other simplifications are also admissible. For example, in case of small module p, all the points of elliptic curve over the field F_p can be easily calculated directly.

Different ways for implementation of computational subsystem are possible. One of the variants was implemented as WEB application. A user's interface for support of elliptic curves computational operations and auxiliary algorithms is presented in Figure 4.

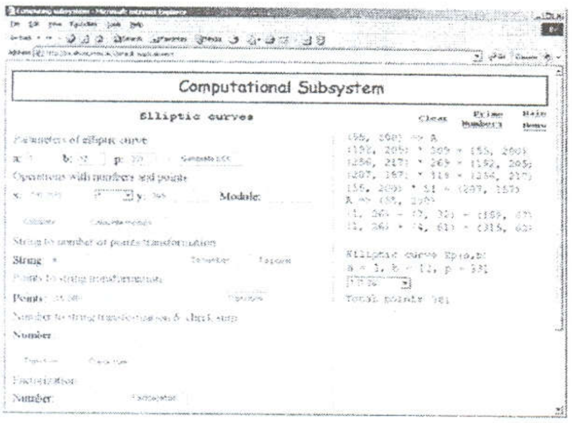
Computational Subsystem
Elliptic curves
Clear
Prime Numbers
Main Menu
Parameters of elliptic curve
Operations with numbers and points
Module:
String to number of points transformation
String
Points to string transformation
Points:
Number to string transformation & check sum
Number
Factorization
Number:

Figure 4. Computational Subsystem

In case of WEB application software is installed in the server side only. Users receive all features via a standard Internet browser application. This concept allows immediate update of subsystem. However the performance of server loading is a critical issue. The local Intranet system copy can be installed for a group of users. Such system was implemented in Moscow Engineering Physics Institute MEPHI (www.mephi.ru). The students have been successfully using it for practical studying of protocols for several years.

4. Trace of Protocol Analysis

Process of experimental protocol analysis can be presented as step-by-step trace. Following is the application example of analysis of confidential message cyclic transferring protocol (described in point 2) with using computational operations and auxiliary algorithms.

System settings:

1. Protocol: Cyclic transferring of confidential message with using RSA
2. Actors: A, B – protocol participants, C – intruder, S – system
3. Attack: read public channel of A and B

Protocol trace is presented below as sequence of interacted processes. Each actor can perform calculations/generation operations or communication events (send/receive messages). System S automatically sends data from public channel of users A and B to intruder C with accordance of attack settings.

Trace:

BEGIN

1.1 A: GENERATE -PARAMS-RSA:

$$\{p = 421, q = 827\}$$

1.2 A: GENERATE-RSA:

$$p = 421$$

$$q = 827$$

$$n = 348167$$

$$phi(n) = 346920$$

$$e_a = 263$$

$$d_a = 105527$$

1.3 A: SEND-PRIVATE: $\{p = 421, q = 827\} \rightarrow$ B

1.4 A: SEND-PUBLIC: $\{e_a = 263, n = 348167\} \rightarrow$ B

4.1 S: SEND: $\{e_a = 263, n = 348167\} \rightarrow$ C

2.1 B: RECEIVE: $\{p = 421, q = 827\}$

2.2 B: RECEIVE: $\{e_a = 263, n = 348167\}$

3.1 C: RECEIVE: $\{e_a = 263, n = 348167\}$

2.3 B: GENERATE-RSA:

$$p = 421$$

$$q = 827$$

$$n = 348167$$

$$phi(n) = 346920$$

$$e_b = 653$$

$$d_b = 145037$$

2.4 B: SEND-PUBLIC: $\{e_b = 653\} \rightarrow$ A

4.2 S: SEND: $\{e_b = 653\} \rightarrow$ C

1.5 A: RECEIVE: $\{e_b = 653\}$

3.2 C: RECEIVE: $\{e_b = 653\}$

1.6 A: GENERATE -TEXT: $M = 2113$

1.7 A: CALCULATE:

$$C_{ab} = 2113^{653} \bmod(348167) = 97682$$

1.8 A: SEND-PUBLIC: $\{C_{ab} = 97682\} \to$ B

4.3 S: SEND: $\{C_{ab} = 97682\} \to$C

2.5 B: RECEIVE: $\{C_{ab} = 97682\}$

3.3 C: RECEIVE: $\{C_{ab} = 97682\}$

2.6 B: CALCULATE:

$$M = 97682^{145037} \bmod(348167) = 2113$$

2.7 B: CALCULATE:

$$C_{ba} = 2113^{263} \bmod(348167) = 330541$$

2.8 B: SEND-PUBLIC: $\{C_{ba} = 330541\} \to$ A

4.4 S: SEND: $\{C_{ba} = 330541\} \to$C

3.4 C: RECEIVE: $\{C_{ba} = 330541\}$

3.5 C: CALCULATE-EUCLIDIAN:

$$[e_a = 263, e_b = 653] \Rightarrow [x = 581, y = -234]$$

3.6 C: CALCULATE-MULTI-INVERSE:

$$[C_{ab} = 97682] \Rightarrow [C_{ab}^{-1} = 263925]$$

3.7 C: CALCULATE:

$$C_{ba}^{x} = 330541^{581} \bmod(348167) = 147442$$

3.8 C: CALCULATE:

$$C_{ab}^{y} = 263925^{234} \bmod(348167) = 347801$$

3.9 C: CALCULATE:

$$M = 147442 \cdot 347801 \bmod(348167) = 2113$$

END

5. Conclusion

The analysis of protocol failures dependent on mathematical properties of used algorithms is a complex problem. The algorithms can be presented as sequences of elementary algebraic operations and auxiliary algorithms. A general set of such operation and algorithms was implemented as a special computational subsystem for experimental performing of algorithms for protocol analysis. Protocol participants and intruders use the computational subsystem for performing required computations. At this point the experimental analysis supplements already existing methods for analysis of security protocols.

References

1. Abadi M., Rogaway P. "Reconciling two views of cryptography (the computational soundness of formal encryption)". *Journal of Cryptology*, 2002; 15(2): 103-127.
2. Backes M., Pitzmann B, Waidner M. "Secure Asynchronous Reactive Systems". *Cryptology ePrint Archive*, Report 2004/082.
3. Dolev D., Yao A. "On the security of public key protocols". *IEEE Transaction on Information Theory*, 1983; 29(2): 198-208.
4. Joye M., Quisquater J. "Cryptanalysis of RSA-type cryptosystems: a visit". *Network Threats*, DIMACS Series in Discr. Math. ant Th. Comp. Sci., AMS, 1998, pp. 21-31.
5. Koyama K., Maurer U.M., Okamoto T., Vanstone S.A. "New public-key schemes based on elliptic curves over the ring Zn". In: Feigenbaum J. (ed) *Advance in Cryptology - Crypto'91*. Springer-Verlag, 1991, pp. 252-266 (Lectures Notes in Computer Science vol. 576).
6. Meadows C. "Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends". *IEEE Journal on Selected Areas in Communication*, 2003; 21(1): 44-54.
7. Mikhailov A.S. "Distributed Information System to Analyze Security Protocols". In: *Proc. of the Workshop on Computer Science and Information Technologies (CSIT'2005)*, Vol. 1. USATU, Ufa, Russia, 2005, pp. 215-218.
8. Moore J. H. "Protocol failures in cryptosystems". IEEE Transaction on Information Theory, vol. 5 T-76, pp. 594-602, 1988
9. Rivest R., Shamir, A., Adleman L. "A method for obtaining digital signatures and public-key cryptosystems". *Communications of the ACM*, 1978; 21(2): 120-126.
10. Simmons G.J. "A 'weak' privacy protocol using the RSA crypto algorithm". *Cryptologia*, 1983; 7(2): 180-182.