On Use of Lightweight Cryptography in Routing Protocols

Sergey Panasenko, Sergey Smagin

Abstract*— *Cryptographic algorithms become more complex and "heavyweight" every year. This is completely correct from the viewpoint of security. But at the same time such growth increases resource requirements of the algorithms and the complexity of their implementation. This also essentially increases expenses of energy required to perform cryptographic procedures.

In this paper we review applications of cryptographic algorithms in routing protocols. Also we analyze the possibilities of use of a lightweight block cipher as a cryptographic kernel to mount various types of cryptographic algorithms which do not require significant resources together over it.

We propose to enlarge the set of cryptographic algorithms required to be implemented within IPsec protocol and to include lightweight encryption and authentication algorithms into the set.

Implementation of lightweight algorithms to apply in IPsec and related network protocols allows to provide adequate moderate security level in various applications where it is not required to use heavy and strong cryptography; it also allows to save energy and reduce the cost of implementation.

Index Terms*— *Lightweight cryptography, KATAN, block cipher, hash function, routing protocol, RIPv2, IPsec.

I. INTRODUCTION

Cryptographic algorithms become more complex and "heavyweight" every year. This is one of the trends of modern cryptography. Block sizes, key lengths, sizes of internal state and other parameters of encryption and authentication algorithms are being increased. This is completely correct from the viewpoint of security. But at the same time such growth increases resource requirements of algorithms and the complexity of their implementation. This also essentially increases expenses of energy required to perform cryptographic procedures.

Information has such properties as its value and actuality. It is not always required to use modern heavy and strong cryptographic algorithms for information protection. Therefore information with one-day lifetime should not be protected by an algorithm with a theoretical time of billions of years to break. The security system should be adequate to the value of protected data.

This principle is correct for systems as a whole; and it also should be applied to systems components, including cryptographic primitives. High levels of security should only be used when they are required.

As a consequence, nowadays as before it is relevant to design effective cryptosystems of moderate security level (e. g. less than 128 bits). This can result in raising requirements to work out some alternative cryptographic standards, which can give users much more flexibility in choosing security levels of systems. In its turn, this allows to save resources, minimize power consumption and so on.

From this point of view it is very promising to use the following approaches:

- lightweight cryptography that tries to find a compromise between low resource requirements, performance and strength of cryptographic primitives [1]; consequently, lightweight algorithms are being developed initially with characteristics required for energy-efficient systems;
- recycling of cryptographic primitives [2], i. e. reusing existing cryptographic primitives or their elements while developing new ones; also recycling means using of various types of cryptographic algorithms based on the same primitive in the same system – it is often required in many applications to use encryption and data authentication at the same time.

In this paper we review applications of cryptographic algorithms in routing protocols. Also we analyze the KATAN lightweight block cipher and the possibilities of its use as a cryptographic kernel to mount various types of cryptographic algorithms which do not require significant resources together over it.

We propose to enlarge the set of cryptographic algorithms required to be implemented within IPsec protocol [3] and to include lightweight encryption and authentication algorithms into the set.

Implementation of lightweight algorithms in IPsec and related network protocols allows:

- to provide adequate moderate security level in various applications where it is not required to use heavy and strong cryptography;
- to save energy and reduce the cost of implementation.

II. CRYPTOGRAPHIC ALGORITHMS IN ROUTING PROTOCOLS USAGE EXAMPLES

Cryptographic algorithms are intensively used in variety of network technologies. Let's review some routing protocols as an example.

Manuscript received on April 14, 2012.

Dr. S. Panasenko, head of software development department, ANCUD Ltd., Moscow, Russia, (e-mail:serg@panasenko.ru).

S. Smagin, senior software developer, ANCUD Ltd., Moscow, Russia, (e-mail: serg@ochacovo.ru).

A. Routing information protocol version 2

Routing information protocol version 2 (RIPv2) was proposed in [4] as a successor of the first version of routing information protocol (RIP) [5].

RIP is one of distance-vector routing protocols. It uses the hop count as a metric of routes. To prevent loops, RIP limits the hop count – the maximum allowed number of hops is 15. Route metric 16 means an infinite distance and is used to mark any undesirable route, which should be excluded from the route selection process. RIP is positioned as a routing protocol for relatively small interior networks.

The main goal of RIPv2 is to provide several extensions to the first version of RIP including the following:

- RIP does not carry any subnet information, i. e. it allows classful routing only; RIPv2 contains the “Subnet Mask” extension so it provides routing in both classful and classless networks;
- RIP does not provide any security information; RIPv2 introduces two extensions to provide authentication: “Authentication Type” and “Authentication”.

The document [4] specifies the only authentication type – the simple password authentication. Therefore “Authentication” extension is able to contain the plain text password with right null bytes padding when required to reach 16 bytes length.

One more authentication type (Keyed Message Digest Algorithm) was proposed by Baker and Atkinson [6] to use the keyed mode of the hash function MD5 [7] as the standard authentication algorithm for RIPv2. Authors of [6] noted that the proposed authentication mechanism was intended to be algorithm-independent, and MD5 hash function could be easily replaced by any other hashing algorithm if MD5 considered to be broken. “Authentication Data” extension (which is used to store MD5 hash value) has variable length to make such replacement possible.

“Authentication” extension defined in [4] was replaced in [6] by the following three extensions with the same size together:

- “RIP-2 Packet Length” extension contains the full length of RIPv2 packet;
- “Key ID” – contains the identifier of preshared key used for authentication;
- “Auth Data Len” – contains the length of “Authentication Data” extension.

Keyed MD5 hashing made the authentication of RIPv2 packets much stronger than initial plaintext password authentication. After MD5 algorithm has considered broken, the document [6] was obsoleted by [8]; the new version of authentication procedure in RIPv2 uses SHA family of hash algorithms (SHA-1, SHA-256, SHA-384, and SHA-512) [9] and the HMAC technique [10] instead keyed MD5 hashing. This should greatly reduce the risk of successful attacks against RIPv2-based routing systems. MD5 hashing remains allowed for backward compatibility only. The structure of RIPv2 packet remains unchanged.

Table 1 - Some data fields of RIPv2 packet

<table>
<tr><td colspan="3">. . .</td></tr>
<tr><td>. . .</td><td colspan="2">Authentication Type</td></tr>
<tr><td>RIPv2 Packet Length</td><td>Key ID</td><td>Auth Data Len</td></tr>
<tr><td colspan="3">Sequence Number</td></tr>
<tr><td colspan="3">. . .</td></tr>
<tr><td colspan="3">Authentication Data</td></tr>
<tr><td colspan="3">. . .</td></tr>
</table>

B. Routing in IPv6 networks

The newest version of routing information protocol – RIP next generation (RIPng) – is used for routing in IPv6 networks. RIPng is an extension of RIPv2 with some differences such as:

- RIPng supports IPv6 networking;
- RIPng does not provide authentication, because IPv6 routers can use IPsec for authentication; consequently, all authentication fields are absent in RIPng packets.

RIPng is described in [11].

As we know, IPsec [3] is a mandatory component of IPv6. IPsec provides security for information transmission over open networks. The aim of IPsec is to acquire data integrity, confidentiality, authentication and protection against replay attacks. IPsec includes the following main subprotocols:

- Authentication Header (AH) [12];
- Encapsulating Security Payloads (ESP) [13].

AH and ESP are used to provide security. ESP must be supported, but support of AH is optional, because ESP can provide sufficient level of security itself. These two protocols can be used individually or in combination [3].

Both AH and ESP contain mandatory “Integrity Check Value” (ICV) field, which is used for data authentication in these protocols. ICV can be calculated with a variety of cryptographic algorithms; some of them are mandatory to be implemented, they are listed in the document [14]. In addition, some optional algorithms can be implemented in IPsec modules.

C. On a lifetime of routing information

Whichever routing protocol is in use, we can see that routing information is updated very frequent. For example, both RIP and RIPv2 protocols update it every 30 seconds (with 180-seconds timeout) [5, 15].

Lifetime of routing information can be supposed a period between updates, and this is the case when we unreasonably use strong cryptography to protect information that remains actual for a very short time period.

Therefore this is the point where we can and must use lightweight cryptography instead of general purpose authentication and encryption algorithms (with a theoretical time of billions of years to break).

III. LIGHTWEIGHT BLOCK CIPHER AS A CRYPTOGRAPHIC KERNEL

Let us take the KATAN block cipher [16] as an example of lightweight block cipher. Also we offer the add-on over KATAN that allows to use it in hashing mode.

A. Brief description of KATAN block cipher

KATAN is a family of three block ciphers with various block sizes: 32, 48, and 64 bits. All the ciphers have 80-bit keys.

Each of KATAN algorithms loads a data block into two internal shift registers $L1$ and $L2$. It performs 254 rounds using nonlinear functions which form the registers feedback (Fig. 1). One of nonlinear functions uses specific irregular value (*IR*) in addition to several register bits. This value depends on the round number.

The resource requirements of KATAN are extremely low because of the following collection of factors:

- KATAN uses shift registers, which can be implemented easily; feedback functions are very simple too, though they provide required nonlinearity;
- it processes small blocks of data – 32 to 64 bits;
- its internal state is small, its size is a little bit greater than the block size.

B. Hashing add-on over KATAN

KATAN block cipher can be used as a cryptographic kernel for mounting other kinds of cryptographic primitives over it. The set of cryptographic functions over KATAN was recently proposed in [17]. This set includes:

- block cipher – KATAN algorithm itself;
- stream cipher and pseudo random number generator – see [17] for details;
- hash function.

To minimize expenses, the hashing add-on should be as lightweight as possible. One of hash functions with a thin hashing layer over the internal block cipher is CRUNCH [18] algorithm, which took part in the first stage of SHA-3 contest. One of CRUNCH versions is based on the double-pipe Merkle-Damgård construction. The double-pipe version allows to reach higher cryptographic strength comparably to the main version with practically the same overheads [19].

Using the compression function structure similar to CRUNCH (strengthened version) and the 64-bit KATAN64 block cipher, we can build a lightweight compression function (Fig. 2).

The compression function of the double-pipe CRUNCH version encrypts every block of the message twice: concatenated with H_i and concatenated with H'_i values [19]. We slightly modified the structure of the CRUNCH compression function: as the block size of the KATAN64 cipher is relatively small, every block of the message is separated into two halves: M'_i and M''_i, which are processed by the block cipher in parallel. Final hash value is a result of the final transformation of H_N and H'_N values (last message block processing output values).

The number of additional GE (gate equivalent) required for implementation of the discussed hash function and stream cipher can be estimated as 800–1000. Thus, the described set of cryptoprimitives (including KATAN64) requires very modest resources – about 2000-2200 GE. This is comparable to most well-known lightweight block ciphers (see e. g. [16]).

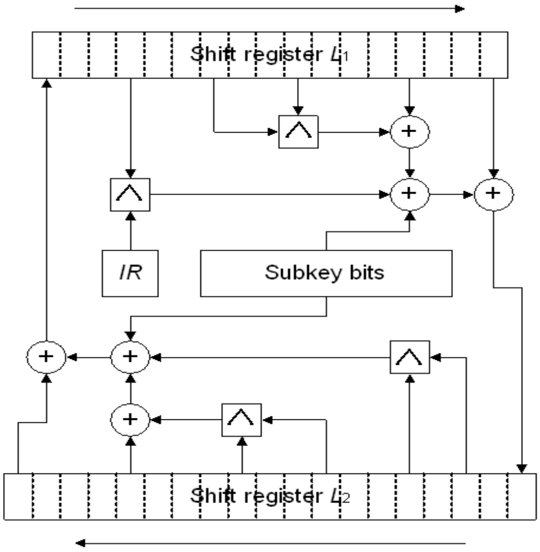

Figure 1 - Round function of KATAN

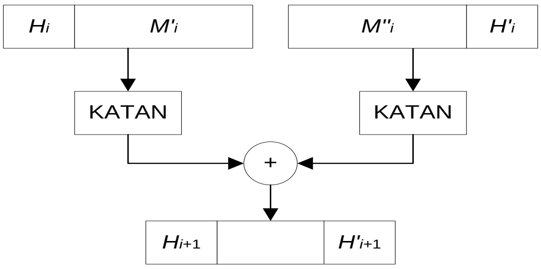

Figure 2 - KATAN64-based compression function

IV. ON USE OF LIGHTWEIGHT CRYPTOGRAPHY IN ROUTING

The document [14] enumerates several authentication and encryption algorithms for use in IPsec subprotocols to provide interoperability of various IPsec implementations.

Tables 2 and 3 contain cryptographic algorithm implementation requirements within ESP [14] (excluding "NULL" algorithms and algorithms that should not be implemented). The key words that determine the levels of requirement are described in [20] and [14].

The list of authentication algorithms for use within AH (defined in [14]) is completely equivalent to the list of ESP authentication algorithms.

All required algorithms have relatively similar characteristics and cryptographic strength. We suppose that such strong algorithms are not required for every IPsec implementation (e. g. when IPsec is used to protect routing information). Lightweight cryptoalgorithms save energy and reduce the cost of implementation. Therefore we suppose that lists of algorithms (to be implemented within ESP and AH) should be revised to permit energy-efficient lightweight algorithms when possible.

The proposed algorithms to add into the recommended set for implementation as encryption and authentication algorithms are given in tables 4 and 5, respectively.

Table 2 - ESP encryption algorithms

Requirement	Algorithm	Reference
MUST	AES-CBC with 128-bit keys	[21]
MUST-	TripleDES-CBC	[22]
SHOULD	AES-CTR	[23]

Table 3 - ESP authentication algorithms

Requirement	Algorithm	Reference
MUST	HMAC-SHA1-96	[24]
SHOULD+	AES-XCBC-MAC-96	[25]
MAY	HMAC-MD5-96	[26]

Table 4 – Proposed ESP encryption algorithm

Requirement	Algorithm
MAY	LWC-CBC

Table 5 – Proposed ESP & AH authentication algorithm

Requirement	Algorithm
MAY	HMAC-LWH

Where:

- "LWC-CBC" means any suitable lightweight block cipher in CBC mode of operation [27], e. g. KATAN block cipher described in section 3.1;
- "HMAC-LWH" is the HMAC [10] construction over a lightweight hashing algorithm, e. g. over KATAN block cipher in hashing mode described in section 3.2.

Such algorithms implementation allows to provide adequate moderate security level in various applications where it is not required to use strong (therefore, heavy) cryptography. This extension does not affect the remaining functionality of IPsec and allows to use benefits of IPsec infrastructure in cooperation with lightweight algorithms.

As it has been shown in section 2.3, IPsec is just an example: similar modifications are actual for other routing protocols, e. g. for RIPv2.

V. CONCLUSION

In this paper we examined cryptographic algorithms used in routing protocols. Also we analyzed recycling possibilities of the KATAN block cipher and its use as a hashing algorithm. We propose to enlarge the set of cryptographic algorithms to be implemented within ESP and AH protocols and to include lightweight encryption and authentication algorithms into the set. Implementation of lightweight algorithms in IPsec and related network protocols allows:

- to provide adequate moderate security level in various applications where it is not required to use strong (therefore, heavy) cryptography;
- to save energy and reduce the cost of implementation.

ACKNOWLEDGMENT

We would like to thank Anton Konvaluk and Denis Tverdunov for reviewing and valuable comments on this paper.

REFERENCES

1. A. Poschmann. Lightweight Cryptography from an Engineers Perspective. Workshop on Elliptic Curve Cryptography (ECC 2007).
2. J. Troutman and V. Rijmen. Green Cryptography: Cleaner Engineering Through Recycling. IEEE Security & Privacy, vol. 7, no. 4, pp. 71-73, July/August, 2009.
3. S. Kent, K. Seo. RFC 4301. Security Architecture for the Internet Protocol. December 2005.
4. G. Malkin. RFC 1388. RIP Version 2. Carrying Additional Information. January 1993.
5. C. Hedrick. RFC 1058. Routing Information Protocol. June 1988.
6. F. Baker, R. Atkinson. RFC 2082. RIP-2 MD5 Authentication. January 1997.
7. R. Rivest. RFC 1321. The MD5 Message-Digest Algorithm. April 1992.
8. R. Atkinson, M. Fanto. RFC 4822. RIPv2 Cryptographic Authentication. February 2007.
9. FIPS PUB 180-2. Secure Hash Standard. National Institute of Standards and Technology, U. S. Department of Commerce – August 2002.
10. H. Krawczyk, M. Bellare, R. Canetti. RFC 2104. HMAC: Keyed-Hashing for Message Authentication. February 1997.
11. G. Malkin, R. Minnear. RFC 2080. RIPng for IPv6. January 1997.
12. S. Kent. RFC 4302. IP Authentication Header. December 2005.
13. S. Kent. RFC 4303. IP Encapsulating Security Payload (ESP). December 2005.
14. V. Manral. RFC 4835. Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH). April 2007.
15. G. Malkin. RFC 2453. RIP Version 2. November 1998.
16. C. De Cannière, O. Dunkelman, M. Knežević. KATAN & KTANTAN – A Family of Small and Efficient Hardware-Oriented Block Ciphers. CHES'09, LNCS, vol. 5747, pp. 272-288. Springer, 2009.
17. S. Panasenko, S. Smagin. Energy-efficient cryptography: application of KATAN. SoftCOM 2011. 19. International Conference on Software, Telecommunications & Computer Networks. Split – Hvar – Dubrovnik, September 15-17, 2011. Proceedings (SS2 – Special Session on Green Networking).
18. J. Patarin, L. Goubin, M. Ivascot, W. Jalby, O. Ly, V. Nachef, J. Treger, E. Volte. CRUNCH. Specification. // Available at http://csrc.nist.gov – October 28, 2008.
19. E. Volte. CRUNCH. A SHA-3 Candidate. // Available at http://www.voltee.com – 27 February 2009.
20. S. Bradner. RFC 2119. Key words for use in RFCs to Indicate Requirement Levels. March 1997.
21. S. Frankel, R. Glenn, S. Kelly. RFC 3602. The AES-CBC Cipher Algorithm and Its Use with IPsec. September 2003.
22. R. Pereira, R. Adams. RFC 2451. The ESP CBC-Mode Cipher Algorithms. November 1998.
23. R. Housley. RFC 3686. Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP). January 2004.
24. C. Madson, R. Glenn. RFC 2404. The Use of HMAC-SHA-1-96 within ESP and AH. November 1998.
25. S. Frankel, H. Herbert. RFC 3566. The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec. September 2003.
26. C. Madson, R. Glenn. RFC 2403. The Use of HMAC-MD5-96 within ESP and AH. November 1998.
27. NIST Special Publication 800-38A. Recommendation for Block Cipher Modes of Operation. Methods and Techniques. National Institute of Standards and Technology, U. S. Department of Commerce – December 2001.

AUTHORS PROFILE

Dr. S. Panasenko received his Ph.D. from Moscow Institute of Electronic Engineering, Russia (2003). Since 1996 he works at ANCUD Ltd. as a software developer and (since 1999) as the head of software development department. He is an author of two books (in Russian) in a field of cryptography. Senior member of IACSIT (2011). His fields of interest include cryptology and security of computer systems and networks.

S. Smagin since 2005 works at ANCUD Ltd. as a senior software developer. His fields of interest include cryptology and security of computer systems and networks.