Int. J. Communications, Network and System Sciences, **2012, 5, 245-252**
http://dx.doi.org/10.4236/ijcns.2012.55033 Published Online May 2012 (http://www.SciRP.org/journal/ijcns)

Corporate Intranet Security: Packet-Level Protocols for Preventing Leakage of Sensitive Information and Assuring Authorized Network Traffic

Boris S. Verkhovsky, Roberto D. Rubino
New Jersey Institute of Technology, University Heights, Newark, USA
Email: {verb, rr8}@njit.edu

Received March 23, 2012; revised April 20, 2012; accepted May 7, 2012

ABSTRACT

Securing large corporate communication networks has become an increasingly difficult task. Sensitive information routinely leaves the company network boundaries and falls into the hands of unauthorized users. New techniques are required in order to classify packets based on user identity in addition to the traditional source and destination host addresses. This paper introduces Gaussian cryptographic techniques and protocols to assist network administrators in the complex task of identifying the originators of data packets on a network and more easily policing their behavior. The paper provides numerical examples that illustrate certain basic ideas.

Keywords: Corporate Security; Authorized Traffic; Data Leakage; Cryptographic Token; Authentication; Trusted Authorities; Toom-Cook Algorithm

1. Introduction

Modern Internet Protocol (IP) networks deployed within large organizations face a multitude of threats, both from within the network borders and from the Internet. Network administrators are inundated with new network security compliance challenges that are mandated by myriad government agencies and industry groups. Security breaches such as data leakage and industrial espionage often originate within the corporate firewall and are therefore difficult to detect. A new system of authentication is needed to enable the network as a whole, along with its administrators, to identify the users, who are generating packets traversing the network. By tagging all IP packets on the corporate local-area network (LAN) with identity information, network devices and administrators can monitor, audit and shape the flow of data using familiar user identities instead of the traditional IP quintuple.

Remark 1: Large corporate networks are made up of multiple local-area networks (LANs) connected together via a wide-area network (WAN). Packets are tagged within each individual LAN and the tag is verified by devices within both the LAN and WAN.

2. Related Work

Packet marking has been used to solve a broad array of problems in the networking space. Quality of Service (QoS) is accomplished in many networks by tagging IP packets with a TOS or DSCP [1] value to indicate service discrimination on the network. By augmenting each IP packet with an additional header, the IPSEC authentication header [2] provides end-to-end packet integrity, authentication and replay protection. The Microsoft CHOICE network [3] was designed to secure wireless Internet access in public places. Their research leveraged packet marking to tag packets with a user identifying cryptographic token. As traffic passes through a central server, the token is examined and used to verify user identity and enforce access control. It is widely accepted that distributed denial-of-service attacks can be mitigated with an efficient mechanism to discover and throttle the source of the attacking packets. Packet marking has been employed as a possible solution to this problem [4-6], these schemes in particular utilize unused or underutilized bits in the IP header to facilitate the traceback algorithm. In order to secure modern military networks, [7] introduces a scheme, which marks IPv6 packets with an extension header containing an elliptic curve digital signature. Hardware contained within all network nodes validates the public-key based signature for authenticity before accepting a packet. In order to facilitate resource access control at the network edge, [8] explores the use of packet marking for authenticating traffic between end users and ISP edge routers. The use of this technique facilitates the secure transmission of real-time data along with securing access to subscription-based content. A proposed vendor-neutral firewall authentication and iden-

tity-based packet filter scheme based on packet marking is discussed in [9]. This scheme introduces identity carrying IPv4 option headers to inform mid-stream firewalls of user identity in order to enhance the standard quintuple-based firewall packet filters. Packet marking is employed to realize path pinning in packet-switched networks [10]. Path pinning allows IP networks to behave in a manner similar to traditional circuit-switched networks. In this paper, we explore using packet marking for end-point-to-network security. In other words, we wish to make the network as a whole aware of user identities in addition to host identities by marking each packet with authentication information. As a result, this information can be interpreted and acted upon by network devices during packet routing.

3. Protocol Description

The goal of the Corporate Intranet Security (CIS) protocol is to insert into each packet an additional header which securely identifies the user responsible for its origination. Special devices on the network can interpret this header and determine the originating user and the authenticity of the packet. These devices can additionally enforce policies (such as packet prioritization and/or auditing) based on the detected identities. A single trusted entity is responsible for all identities existing on the network and has the ability to share this information with the appropriate security devices. **Figure 1** illustrates such a network. An analogous scheme introduced in [7] relies on public-key cryptography and specialized hardware installed on all network nodes. Our scheme proposes symmetric encryption in order to eliminate the need for dedicated cryptographic coprocessors.

3.1. Major Participants and Components

Network Users: Each network user is identified by the appropriate subscript index $j = 1, 2, \cdots, n$. Every user has an associated user name u_j (such as u_1 = Alice and u_2 = Bob), password ρ_j and randomly assigned temporary user identifier (*uid*) w_j.

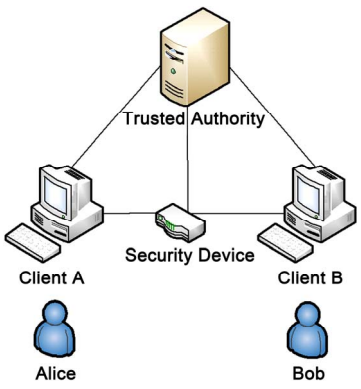

Figure 1. Network overview.

Corporate Traffic Controller (CTC): CTCs can be any network device that has the requisite processing power to validate user identities on a per-packet basis. Additionally, CTCs can influence the flow of traffic within the network (*i.e.* firewalls, routers, packet shapers, etc). Similar to network users, CTCs are identified by a subscript index $t = 1, 2, \cdots, h$ ($h \ll n$) along with an associated username v_t and password π_t.

Trusted Authority (TA): TAs are responsible for the authentication of users on the network and the generation of the associated cryptographic keys. They are also responsible for sharing identity information with the appropriate CTCs. For redundancy purposes, there may be multiple TAs. However for the sake of simplicity, we consider only one.

Authentication Key (AK): When a user or CTC transmits its private information to the TA, it is desirable for this information to be encrypted. To that end, the AK is used to encrypt the channel between u_j / v_t and the TA. The joint AK between the user u_j and the TA is a Gaussian integer denoted:

$$K_j := \{e_j, f_j\} \ \{j = 1, \cdots, n\} \tag{1}$$

The joint AK between the CTC and the TA is denoted:

$$K_{-t} := \{e_{-t}, f_{-t}\} \ \{t = 1, \cdots, h\} \tag{2}$$

User Digital Signature: Each user is required to digitally sign the contents of their packets and embed such signature within the packet. While there are a multitude of methods for introducing a user's signature [11-13], our primary concern in choosing a digital-signature scheme is to minimize the processing time-delay required for its verification at the CTC. Before creating the digital signature, a predetermined randomly generated set of bytes are appended to the packet (*i.e.* salted) in order to provide greater crypto-immunity. This salt is denoted: ς_j.

User Digital Signature Key: The TA selects a key for each user u_j in order to create the user digital signature. The digital signature key for user u_j is another Gaussian integer denoted:

$$L_j := \{q_j, r_j\} \ \{j = 1, \cdots, n\} \tag{3}$$

Each digital signature key L_j is valid for a set period of time and has an associated expiration time ε_j.

3.2. Protocol Overview

User Actions: In order to embed the necessary identity information into every packet, each user u_j must execute the following steps:

1) User Authentication Exchange—a process in which user u_j proves its identity to the TA and obtains the necessary data required to communicate on the network.

2) Digital Signature Process—the algorithm used to create a digital signature for each outbound packet.
3) Packet Marking—the process of embedding a digital signature into each outbound packet.

CTC Actions: The CTC must also execute a series of steps in order to perform its duties:

1) CTC Authentication Exchange—a process in which the user that represents a CTC v_t proves its identity to the TA and obtains the necessary data to validate digitally signed packets traversing the network.
2) Identity Verification & Policy Enforcement—the process which the CTC uses to inspect, validate and optionally execute policies against arriving packets.

3.3. User Authentication Exchange (UAX)

A user u_j must identify itself to the TA before it can participate on the network. The authentication process is as follows:

1.1. User u_j requests a UAX from the TA.

2.1. The TA selects a large prime *p* and a pair of distinct positive integers $g := \{c, d\}$ for which the inequality

$$\left(c^2 + d^2\right) \bmod p \neq 0 \text{ holds.} \tag{4}$$

Remark 2: If *p* mod 4 = 3, then (4) automatically holds for every Gaussian integer.

3.1. The TA transmits *p* and *g* to the user u_j.

4.1. The user u_j randomly selects a large secret integer b_j satisfying the inequality

$$2 \le b_j \le p - 2. \tag{5}$$

5.1. Correspondingly, the TA also selects a large secret integer b_0 on the same interval:

$$2 \le b_0 \le p - 2. \tag{6}$$

6.1. User u_j computes

$$x_j := (c, d)^{b_j} \bmod p \tag{7}$$

and transmits x_j to the TA.

7.1. The TA computes

$$y := (c, d)^{b_0} \bmod p \tag{8}$$

and transmits *y* to user u_j.

8.1. User u_j computes

$$K_{0j} := y^{b_j} \bmod p \tag{9}$$

9.1. The TA computes

$$K_{j0} := x_j^{b_0} \bmod p \tag{10}$$

It is easy to verify that the equation

$$K_{j0} = K_{0j} \tag{11}$$

holds for every $j = \{1, 2, \cdots, n\}$.

Let

$$K_j := K_{j0} := (e_j, f_j) \tag{12}$$

where K_j is the AK between user u_j and the TA.

10.1. User u_j applies its AK K_j to encrypt its username and password and transmits them to the TA:

$$(u_j, \rho_j)_{K_j} \tag{13}$$

11.1. Upon verification of (13), the TA generates the following:

1) w_j—a randomly generated unique 16-bit value representing user u_j. This number cannot be reused until the time ε_j is reached.
2) ς_j—a randomly generated 32-bit value that is appended to each packet in order to strengthen the digital signature.
3) $L_j := \{q_j, r_j\}$—the digital signature key used to construct the digital signature.
4) ε_j—the coordinated universal time (UTC) for which the above values cease to be valid on the network.

12.1. The TA, using the negotiated authentication key K_j encrypts the above quadruple and transmits it to the user u_j:

$$(w_j, \varsigma_j, L_j, \varepsilon_j)_{K_j} \tag{14}$$

Once the UAX process is complete, the user has all of the information necessary to begin the *Digital Signature Construction* and *Packet Marking* processes. Furthermore, once time ε_j is reached, user u_j must only execute steps 10.1-12.1 in order to renew its ability to use the network.

3.4. CTC Authentication Exchange (CAX)

A CTC$_t$ must authenticate with the TA in a similar manner as network users do. In fact, steps 1.1-10.1 of the UAX process are analogously replicated in the CAX process, however the CTC submits its username v_t and password π_t. The new additional steps are as follows:

11.2. Upon verification of (13) and identifying the user v_t as a CTC, the TA, using the negotiated authentication key K_{-t} encrypts the quadruple $(w_j, \varsigma_j, L_j, \varepsilon_j)_{K_{-t}}$ for all j and transmits them to the CTC$_t$.

12.2. As new users register with the TA via the UAX process, the TA must transmit these values to the CTC$_t$. Furthermore, when time ε_j arrives for all j in the CTC's memory, all values with the associated subscript j are to be purged.

After completing the final steps, the CTCs have the information necessary to validate the user identities of arriving packets.

3.5. Digital Signature

A digital signature scheme for the application described

in this paper must meet a number of criteria in order to be effective:

1) Speed—signature construction and verification must be performed as quickly as possible in order to minimize packet transmission delay.

2) Security—the signature must provide adequate security to prevent falsification of the packet header. However, due to frequent key rotations, strength is secondary to speed.

Remark 3: *There is a tradeoff between the level of security and speed of signature generation*. *We have intentionally relaxed the level of crypto-immunity for the sake of decreasing signature generation time*.

3) Size—the signature must not take excessive space in order to minimize the overhead within the packet.

4) Consideration of mutable fields—the internet protocol (IP) header consists of a number of fields:

- Fields that can be modified while the packet is in transit are referred to as mutable fields;
- Fields that remain constant throughout the transmission process are known as non-mutable fields.

An effective packet digital signature algorithm must ignore the mutable fields within the IP header and process the non-mutable fields; otherwise the signature will be invalidated.

3.6. Digital Signature Process

The signature generation process (see **Figure 2**) consists of the following:

1) The user digital signature key $L_j := \{q_j, r_j\}$.

2) An MD5 [14,15] hash denoted as m. The MD5 algorithm takes as input a series of bytes of arbitrary length and produces as output a 128-bit hash. The hash m is constructed over the concatenation of non-mutable IP header fields denoted as β, packet payload denoted as γ and the salt is ς_j.

3) $$m = \mathrm{MD5}(\beta, \gamma, \varsigma_j) \quad (15)$$

4) A signature s is computed as follows:

$$s := (q_j m + r_j) \bmod p \quad (16)$$

5) The signature s is truncated by taking the most significant 96-bits and discarding the rest. This is done to satisfy space constraint requirements and is similar to the process specified in [16].

3.7. Packet Marking

Once the user u_j has obtained the quadruple of necessary parameters $(w_j, \varsigma_j, L_j, \varepsilon_j)$, it can begin marking outgoing packets with the security option header. A packet signature is calculated using the algorithm described above and embedded into the security option header; this header is in turn embedded within the packet.

The security option header (see **Figure 3**) consists of four fields:

1) Code - required by the IP protocol specification, it identifies the option type and contains flags that instruct routers how to process the option.

2) Length - specifies the size of the entire option header.

3) w_j - the temporary user id value of the user u_j that generated this packet. The value w_j is determined during the user authentication exchange.

4) User Digital Signature—the output of the Digital Signature process as described above.

Once constructed, the security option header is placed within the IP packet, as shown in **Figure 4**, and transmitted.

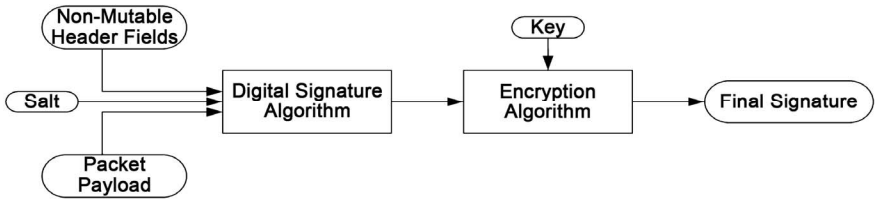

Figure 2. Packet signing process.

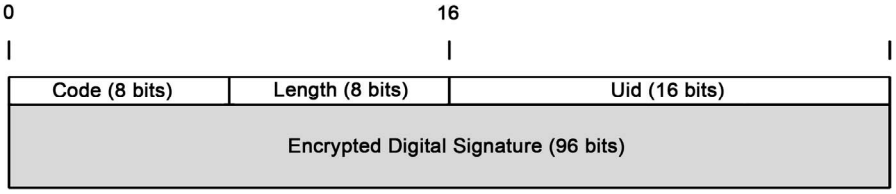

Figure 3. Security option header.

Figure 4. Option header placement.

3.8. Identity Verification & Policy Enforcement

As packets traverse the network, they will be processed by one or more CTCs. The role of these devices is to verify the authenticity of each packet and enforce policies based on the outcome. This behavior is further demonstrated in **Figure 5**. For instance, a corporation may implement a quality of service (QoS) rule that prioritizes all traffic originating from corporate executives. As packets arrive at a CTC, those that are successfully verified as originating from an executive will be prioritized and have their distributed services code point (DSCP) fields updated to reflect their higher priority. Once a CTC performs a CAX, it is provided with the quadruple $\left(w_j, \varsigma_j, L_j, \varepsilon_j\right)$ for all users. Upon arrival, the w_j value in each packet is used to search the CTC's identity table for the corresponding digital signature key L_j and salt ς_j. A new digital signature is calculated and compared with the signature within the packet. If the values match, the associated policies are then executed against the packet. If the values do not match, the packet can either be dropped or stripped of its option header and transmitted according to policies for unsigned packets. The verification procedure is as follows:

1) The CTC examines the arrived packet ϕ and determines the value w_j.

2) The CTC searches its identity table for the digital signature key L_j, salt ς_j and user name u_j that corresponds with w_j.

3) The CTC determines the values of β and γ for the packet ϕ and computes: $m = \mathrm{MD5}\left(\beta, \gamma, \varsigma_j\right)$.

4) The CTC then computes: $\tau := \left(q_j m + r_j\right) \bmod p$.

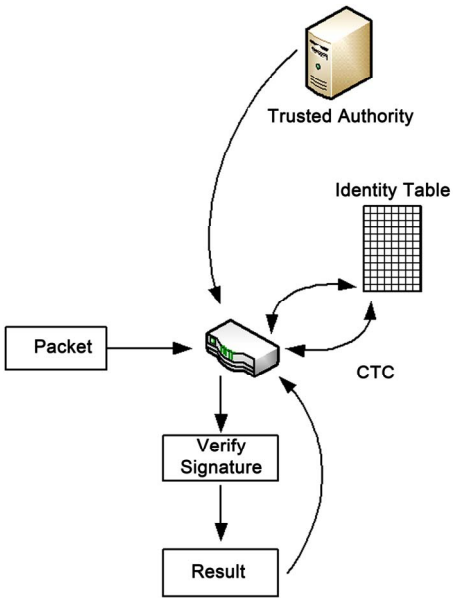

Figure 5. CTC/TA interaction.

5) If $\tau = s$, then the CTC confirms the packet originated from user u_j and executes the associated policy rules.

6) If $\tau \neq s$, then the CTC detects a possible forgery and either drops the packet, strips the option header or executes other associated policy rules.

4. Arithmetic of Gaussian Integers

Definition. Let (*a*,*b*) and (*c*,*d*) be Gaussian integers. Then

$$\left[(a,b) \pm (c,d)\right] \bmod p = (a \pm c, b \pm d) \bmod p; \quad (17)$$

$$\left[(a,b)(c,d)\right] \bmod p = (ac - bd, ad + bc) \bmod p; \quad (18)$$

where the multiplication can be performed faster.

Karatsuba-Ofman algorithm: Let

$$(x,y) := \left[(a,b)(c,d)\right] \bmod p \quad (19)$$

Compute $$P_1 := ac \bmod p; \quad (20)$$

$$P_2 := bd \bmod p; \quad (21)$$

$$P_3 := (a+b)(c+d) \bmod p \quad (22)$$

Then $$x := (P_1 - P_2) \bmod p; \quad (23)$$

$$y := (P_3 - P_1 - P_2) \bmod p \quad (24)$$

Therefore, multiplication of two Gaussians can be performed using three, rather than four multiplications. Hence, the so-called traditional multiplication of complex numbers requires 33.3% more time than the method provided in Karatsuba-Ofman algorithm [17]. In addition, squaring of a Gaussian integer requires two rather than three multiplications. Indeed,

$$(a,b)^2 \equiv \left(a^2 - b^2, 2ab\right) \equiv \left((a-b)(a+b), 2ab\right) \pmod p.$$

In many cases, an increase in the crypto-immunity of an algorithm requires a corresponding increase in the size of the integers used. For extremely large integers that exceed the capabilities of the host computer, the algorithm proposed by Andrei Toom [18] is instrumental. For illustration, an algorithm for multiplication of triple-long integers (e.g. integers three times the size that the host computer can handle) is presented in the appendix.

5. Numeric Illustration-1

Suppose that TA (called *Tom*) selected p = 283 and a generator $g := (c,d) = (1,2)$, and transmitted these values to users *A* and *B* (called *Alice* and *Bob*) and to CTC (called *Clair*). Suppose now *Alice* selects a secret integer a = 51; *Bob* selects a secret integer b = 119; Clair/CTC selects a secret integer c = 257; and *Tom* selects a secret integer t = 171. Each entity performs the following functions (solutions are found in **Table 1**):

Table 1. p = 283; g = (c,d) = (1,2).

	Secret integers	1st stage	2nd stage: Secret keys
Tom (TA)	171	(21, 91)	$(57, 108)_A$; $(122, 212)_B$; $(114, 51)_C$
Alice (A)	51	(230, 107)	$(57, 108)_T$
Bob (B)	119	(38, 21)	$(122, 212)_T$
Claire (CTC)	257	(147, 209)	$(114, 51)_T$

1) Every user computes $x_j \coloneqq (c,d)^{b_j} \bmod p$; $\{j = 1,\cdots,n\}$ and sends x_j to the TA:
   $x_{\text{Alice}} \coloneqq (1,2)^{51} \bmod 283 = (230,107)$;
   $x_{\text{Bob}} \coloneqq (1,2)^{119} \bmod 283 = (38,21)$;
2) The TA computes $y \coloneqq (c,d)^{b_0} \bmod p$ and transmits y to all corporate users and to the CTC:
   $y \coloneqq (1,2)^{171} \bmod 283 = (21,91)$;
3) The CTC computes $x_{-1} \coloneqq (c,d)^{b_{-1}} \bmod p$ and sends x_{-1} to the TA:
   $x_{\text{Claire}} \coloneqq (1,2)^{257} \bmod 283 = (147,209)$;
4) The TA computes $K_{j0} \coloneqq x_j^{\,b_0} \bmod p$ for every $j = \{-1,1,2,\cdots,n\}$:
   $K_{\text{Alice0}} \coloneqq x_{\text{Alice}}{}^{171} \bmod 283 = (57,108)$;
   $K_{\text{Bob0}} \coloneqq x_{\text{Bob}}{}^{171} \bmod 283 = (122,212)$;
   $K_{\text{Claire0}} \coloneqq x_{\text{Claire}}{}^{171} \bmod 283 = (114,51)$;
5) Every user computes $K_{0j} \coloneqq y^{b_j} \bmod p$:
   $K_{0\text{Alice}} \coloneqq y^{51} \bmod 283 = (57,108)$;
   $K_{0\text{Bob}} \coloneqq y^{119} \bmod 283 = (122,212)$;
6) The CTC computes: $K_{0,-1} \coloneqq y^{b_{-1}} \bmod p$:
   $K_{0\text{Claire}} \coloneqq y^{257} \bmod 283 = (114,51)$.

For the sake of brevity, we will not elaborate on the additional computations required for this protocol. However, this will be addressed in a future publication.

6. Prototype Results

A prototype of the system discussed in this paper was developed in C as a suite of device drivers compatible with Microsoft Windows Vista/Server 2008 and higher systems. On desktop systems, packet marking is handled by a lightweight filter (LWF) driver. On Windows-based CTCs, identity verification & policy enforcement is handled by a Windows filtering platform (WFP) driver. These drivers work in conjunction with user-mode tools and a Linux-based trusted authority which together handle the authentication exchange procedures. On Windows PCs equipped with gigabit Ethernet connectivity and dual-core Intel-based processors, the performance impact is negligible. In order to test the CIS system's impact on system performance, large files of approximately 1.3 gigabytes were transferred between two end hosts (Alice & Bob) through a CTC (Claire). **Table 2** contains the results of these tests.

Table 2. Performance results.

	Avg. Throughput	Avg. CPU Utilization	Avg. Time
Without CIS	645 Mbit/s	11.76%	16.645 s
With CIS	492 Mbit/s	32.69%	21.809 s

7. Conclusion

In this paper, we present a technique for transparently identifying users transmitting on a network via packet marking and packet inspection. We also demonstrate a key exchange and digital signature algorithm based on Gaussian integers. By utilizing Gaussian integers for cryptography, we can maintain complexity similar to integer-based schemes while using much smaller prime numbers. Prototypes of the packet marking and inspection components running on dual-processor computers show a modest impact in throughput and CPU utilization. As quad-core and hex-core processors become more popular in the desktop PC space, performance will continue to improve. The preliminary version [19] of this paper was published in proceedings of the 18th International Conference on Software Engineering and Data Engineering.

8. Future Work

Most corporate networks utilize the current generation internet protocol (IPv4) as the network-layer protocol of choice. However, deployment of the next-generation internet protocol (IPv6) within corporations is gaining momentum. The security scheme described in this paper, while based on the IPv4 option header, can be redesigned as an IPv6 extension header. In a future revision, this scheme will be extended to accommodate networks that utilize IPv4, IPv6 or both.

9. Appreciations

The authors wish to express their appreciations to J. Geller, W. Gruver, Yu. Polyakov and I. Semushin for their constructive comments and suggestions that improved the quality of this paper.

REFERENCES

[1] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," *Request for Comments* 2474, 1998.

[2] S. Kent and R. Atkinson, "IP Authentication Header,"

Request for Comments 2401, 1998.

[3] V. Bahl, A. Balachandran and S. Venkatachary, "The CHOICE Network: Broadband Wireless Internet Access in Public Places," Microsoft Research, 2000.

[4] M. Adler, "Trade-Offs in Probabilistic Packet Marking for IP Trace-Back," *Journal of the ACM*, Vol. 52, No. 2, 2005, pp. 217-244. doi:10.1145/1059513.1059517

[5] M. T. Goodrich, "Efficient Packet Marking for Large-Scale IP Traceback," *Proceedings of the* 9*th ACM Conference on Computer and Communications Security*, Washington, 27-30 October 2003, pp. 117-126.

[6] M. T. Goodrich, "Probabilistic Packet Marking for Large-Scale IP Traceback," *IEEE/ACM Transactions on Networking*, Vol. 16, No. 1, 2008, pp. 15-24. doi:10.1109/TNET.2007.910594

[7] C. Candolin, J. Lundberg and H. Kari, "Packet Level Authentication in Military Networks," 6*th Australian Information Warfare & IT Security Conference*, Geelong, 24-25 November 2005.

[8] H. Wang, A. Bose, M. El-Gendy and S. Kang, "IP Easypass: A Light-Weight Network-Edge Resource Access Control," *IEEE/ACM Transactions on Networking*, Vol. 13, No. 6, 2005, pp. 1247-1260. doi:10.1109/TNET.2005.860113

[9] R. D. Rubino, "An Open System for Transparent Firewall Authentication and User Traffic Identification within Corporate Intranets," *Proceedings of the* 9*th ACM SIGITE Conference on Information Technology Education*, Cincinnati, 16-18 October 2008, pp. 113-118. doi:10.1145/1414558.1414591

[10] B. Parno, A. Perrig and D. Anderson, "SNAPP: Stateless Network-Authenticated Path Pinning," *Proceedings of the* 2008 *ACM Symposium on Information, Computer and Communications Security*, Tokyo, 14 May 2008, pp. 168-178.

[11] B. Schneier, "Secrets and Lies: Digital Security in a Network World," 1st Edition, John Wiley & Sons, New York, 2000.

[12] B. Verkhovsky, "Fast Digital Signature Hybrid Algorithm Based on Discrete Logarithm, Entanglements of Plaintext Arrays and Factorization," In: Y. Polyanskov, Ed., 7*th International Conference Mathematics Modeling in Physics, Technology, Socio-Economic Systems and Proc- esses*, Ulyanovsk University, Ulyanovsk, 2009.

[13] B. Verkhovsky, "Overpass-Crossing Scheme for Digital Signature," *Proceedings of* 13*th International Conference on Systems Research, Informatics and Cybernetics*, Baden-Baden, 30 July-3 August 2001.

[14] R. Rivest, "The MD5 Message-Digest Algorithm," *Request for Comments* 1321, 1992.

[15] B. Schneier, "Applied Cryptography: Protocols, Algorithms and Source Code in C," 2nd Edition, John Wiley and Sons, New York, 1996.

[16] C. Madson and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH," *Request for Comments* 2403, 1998.

[17] A. Karatsuba and Yu. Ofman, "Multiplication of Multi-Digit Numbers on Automata," *Soviet Physics-Doklady*, Vol. 7, 1963, pp. 595-596.

[18] A. Toom, "The Complexity of a Scheme of Functional Elements Realizingthe Multiplication of Integers," *Soviet Mathematics-Doklady*, Vol. 7, 1963, pp. 714-716.

[19] B. S. Verkhovsky and R. D. Rubino, "Internal Corporative Security: Protocols Preventing Leakage of Sensitive Information and Assuring Authorized Network Traffic in the Domain of Gaussian Integers," *Proceeding of the* 18*th International Conference on Software Engineering and Data Engineering*, Las Vegas, 22-24 June 2009, pp. 244-249.

[20] R. Crandall and C. Pomerance, "Prime Numbers: Computational Perspective," Springer, New York, 2001.

APPENDIX

A1. Multiplication of Triple-Long Integers

Goal: To compute $C = AB$,

where $A = 10^{2n}a_2 + 10^n a_1 + a_0$;

and $B = 10^{2n}$

$b_2 + 10^n b_1 + b_0$;

Inputs: $a_2,a_1,a_0,b_2,b_1,b_0,n$.

Consider $A(x):=a_2x^2+a_1x+a_0$;

and $B(x) = b_2x^2 + b_1x + b_0$;

Then $C(x) = A(x)B(x)$

$= (a_2x^2 + a_1x + a_0)(b_2x^2 + b_1x + b_0)$

$= c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c$.

Outputs: c_4,c_3,c_2,c_1,c_0; {if we know them, and x: $= 10^n$ we can compute the product AB}.

Efficiency requirement: minimal number of multiplications.

A2. The Algorithm

Step 1: Compute five products $P_0, \cdots, P_4$:

$P_0 = a_0b_0 = c_0$;

$P_1 = (a_2 + a_1 + a_0)(b_2 + b_1 + b_0)$;

$P_2 = (a_2 - a_1 + a_0)(b_2 - b_1 + b_0)$;

$P_3 = (4a_2 + 2a_1 + a_0)(4b_2 + 2b_1 + b_0)$;

$P_4 = (4a_2 - 2a_1 + a_0)(4b_2 - 2b_1 + b_0)$;

Step 2: Compute

$D_1:= (P_1 - P_2)/2$; $S_1:=(P_1 + P_2)/2$;

$D_2:= (P_3 - P_4)/4$; $S2:=(P_3 + P_4)/8$;

Step 3: Compute $c_3 = (D_2 - D_1)/3$;

$c_1 = D_1 - c_3$;

Step 4: Compute $c_4 = (S_2 - S_1)/3 + c_0/4$;

$c_2 = S_1 - c_0 - c_4$;

Step 5: $C = 10^{4n}c_4 + 10^{3n}c_3 + 10^{2n}c_2 + 10^n c_1 + c_0$.

A3. Algorithm Validation

$$P_0 = C(0) = a_0b_0 = c_0; \quad \text{(A0)}$$

$$P_1 = C(1) = (a_2 + a_1 + a_0)(b_2 + b_1 + b_0) = c_4+c_3+c_2+c_1+c_0; \quad \text{(A1)}$$

$$P_2 = C(-1) = (a_2 - a_1 + a_0)(b_2 - b_1 + b_0) = c_4 - c_3 + c_2 - c_1 + c_0; \quad \text{(A2)}$$

$$P_3 = C(2) = (4a_2 + 2a_1 + a_0)(4b_2 + 2b_1 + b_0) = 16c_4 + 8c_3 + 4c_2 + 2c_1 + c_0; \quad \text{(A3)}$$

$$P_4 = C(-2) = (4a_2 - 2a_1 + a_0)(4b_2 - 2b_1 + b_0) = 16c_4 - 8c_3 + 4c_2 - 2c_1 + c_0; \quad \text{(A4)}$$

We have four simple linear equations with four unknowns

$$c_4, c_3, c_2, c_1, \text{ and } c_0 := a_0b_0.$$

Let's demonstrate how to find c_4, c_3, c_2, c_1 if we know $P_0, P_1, \cdots, P_4$.

Subtract (A2) from (A1):

$$D_1 := (P_1 - P_2)/2 = c_3 + c_1; \quad \text{(A5)}$$

Subtract (A4) from (A3):

$$(P_3 - P_4)/4 = 4c_3 + c_1; \quad \text{(A6)}$$

Solve two equations with two unknown c_3 and c_1:

Subtract (A5) from (A6):

$$c_3 = [(P_3 - P_4)/4 - D_1]/3; \quad \text{(A7)}$$

$$c_1 = (P_1 - P_2)/2 - c_3. \quad \text{(A8)}$$

We derive for c_2 and c_4 an analogous system of linear equations. Indeed,

Add (A1) and (A2):

$$S_1 := (P_1 + P_2)/2 - c_0 = c_4 + c_2; \quad \text{(A9)}$$

Add (A3) and (A4):

$$(P_3 + P_4 - 2c_0)/8 = 4c_4 + c_2. \quad \text{(A10)}$$

Solve two equations with two unknown c_4 and c_2.

Subtract (A9) from (A10):

$$c_4 = [(P_3 + P_4 - 2c_0)/8 - S_1]/3; \quad \text{(A11)}$$

$$c_2 = (P_1 + P_2)/2 - c_0 - c_4. \quad \text{(A12)}$$

Therefore, (A1)-(A12) describe an algorithm for the multiplication of triple-long integers that uses five products instead of nine.

Compute

$$D_2 := (P_3 - P_4)/4 \text{ and } S_2 := (P_3 + P_4)/8. \quad \text{(A13)}$$

A4. Integrality of Coefficients $c_0, c_1, \cdots, c_4$

PropositionA1: Every coefficient $c_0, c_1, \cdots, c_4$ is an integer if every a_k and b_k is an integer.

Proof: Let us show that c_3 in (A7) is an integer. First of all, (A5) and (A6) are integers, since definitions (A1) and (A2) imply that $(P_1 - P_2)\text{mod}2 = 0$ and, analogously, definitions (A3) and (A4) imply that $(P_3 - P_4)\text{mod}4 = 0$.

Indeed, in the latter case:

$$\begin{aligned}&(P_3 - P_4)\text{mod}4\\ &= [(4a_2 + 2a_1 + a_0)(4b_2 + 2b_1 + b_0)\\ &\quad - (4a_2 - 2a_1 + a_0)(4b_2 - 2b_1 + b_0)]\text{mod}4\\ &= [(2a_1 + a_0)(2b_1 + b_0)\\ &\quad - (-2a_1 + a_0)(-2b_1 + b_0)]\ \text{mod}4\\ &= [(2a_1 + a_0)(2b_1 + b_0) - (2a_1 + a_0)(2b_1 + b_0)]\\ &= 0.\end{aligned}$$

On the other hand, since $gcd(3,4) = 1$, in a proof that c_3 is an integer it is sufficient to show that $4c_3\text{mod}3 = 0$.

Indeed,

$$\begin{aligned}4c_3\text{mod}3 &= [(P_3 - P_4) - 2(P_1 - P_2)]\text{mod}3\\ &= [(a_2 + 2a_1 + a_0)(b_2 + 2b_1 + b_0)\\ &\quad - (a_2 + a_1 + a_0)(b_2 + b_1 + b_0)\\ &\quad - 2(a_2 + a_1 + a_0)(b_2 + b_1 + b_0)\\ &\quad + 2(a_2 + 2a_1 + a_0)(b_2 + 2b_1 + b_0)]\text{mod}3\\ &= 0;\end{aligned}$$

therefore, $4c_3\text{mod}3 = 0$.

Analogously, a reader of this paper can demonstrate that c_4 is also an integer.

Q.E.D.

A5. Numeric Illustration

Let $A = 311593$ and $B = 271628$; compute $C = AB$.

Now let $a_2 := 31$; $a_1 := 15$; $a_0 := 93$; $b_2 := 27$; $b_1 := 16$; $b_0 := 28$ and $x := 100$.

Then $c_0 = P_0 = a_0b_0 = 93 \times 28 = 2604$;

$P_1 = (31 + 15 + 93)(27 + 16 + 28) = 139 \times 71 = 9869$;

$P_2 = (31 - 15 + 93)(27 - 16 + 28) = 109 \times 39 = 4251$;

$P_3 = (4 \times 31 + 2 \times 15 + 93)(4 \times 27 + 2 \times 16 + 28) = 247 \times 168 = 41496$;

$P_4 = (4 \times 31 - 2 \times 15 + 93)(4 \times 27 - 2 \times 16 + 28) = 187 \times 104 = 19448$;

$D_1 = (9869 - 4251)/2 = 2809$;

$S_1 = (9869 + 4251)/2 = 7060$;

$D_2 = (41496 - 19448)/4 = 5512$;

$S_2 = (41496 + 19448)/8 = 7618$;

$c_3 = (5512 - 2809)/3 = 901$;

$c_1 = (2809 - 901) = 1908$;

$c_4 = (7618 - 7060)/3 + 2604/4 = 186 + 651 = 837$;

$c_2 = 7060 - 2604 - 837 = 3619$;

$$\begin{aligned}C &= 100^4 \times 837 + 100^3 \times 901 + 100^2 \times 3619 + 100 \times 1908 + 2604\\ &= 83700000000 + 901000000 + 36190000 + 190800 + 2604 = 84637383404\\ &= AB = 311593 \times 271628.\end{aligned}$$

Computational complexity of Toom's algorithm is discussed in [20].

