Detección de Anomalías en Oráculos Criptográficos tipo RSA por medio de análisis probabilísticas y estadísticos.

Ing. Antonio CASTRO LECHTALER[1] Msc.
Esp. Marcelo CIPRIANO[2]
Eduardo MALVACIO[3]

Laboratorio de Investigación en Técnicas Criptográficas y Seguridad Teleinformática.
Escuela Superior Técnica "Gral Div. Manuel N. Savio".
Facultad de Ingeniería.
Instituto de Educación Superior del Ejército Argentino.

[1]acastro@iese.edu.ar
[2]marcelocipriano@iese.edu.ar
edumalvacio@gmail.com

1. Resumen.

Esta línea de investigación persigue la elaboración de herramientas matemáticas susceptibles de ser sistematizadas en un software capaz de detectar anomalías y mal funcionamiento en servicios de infraestructura de clave pública (PKI) que utilicen el esquema RSA.

Esta herramienta se puede aplicar en redes Públicas o Privados, Lan´s, o Wan´s e incluso Internet; en sistemas militares como del ámbito civil.

Dada la complejidad de los sistemas actuales, se hace cada vez más complicada la detección de determinado tipos de errores. Los mismos pasan de las versiones de pruebas a las versiones definitivas. Dichos errores provocan debilitamiento de los sistemas que la PKI intenta proteger. Incluso se podría sospechar la inoculación intencional de estos bugs, aunque no en todos los casos [01].

El esquema RSA utiliza números primos grandes para los actuales avances en la factorización de números: 1024, 2048 e inclusive 4096 bits de longitud o mayores aún. Un sesgo en la selección de estos valores primos podría provocar una brecha en la seguridad.

Llamaremos Oráculo al un servicio que ofrezca a sus clientes claves públicas, privadas y módulos en un esquema de cifrado/descifrado RSA[1] en una Infraestructura de Clave Pública.

Esta herramienta informática que estamos elaborando permitiría la detección de mal funcionamiento en la distribución probabilística de los factores primos. Sin embargo dada la astronómica cantidad de números posibles, la detección es estadística, mediante el estudio de muestras suministradas por el Oráculo en observación.

2. Palabras Claves.

Seguridad en Redes, Infraestructura de Clave Pública, PKI, Detección de Anomalías, Open-SSL, RSA.

[1] La 3-tupla (*n,e,d*): n (módulo) es el producto de 2 primos, e (clave pública) y d (clave privada) son inversos entre sí mód $\phi(n)$.

3. Contexto.

El Laboratorio de Investigación en Técnicas Criptográficas y Seguridad Teleinformática pertenece a la Escuela Superior Técnica –Facultad de Ingeniería- del Ejército Argentino en el área del Posgrado en Criptografía y Seguridad Informática que se dicta en esta institución, junto a otros posgrados y carreras de grado en ingeniería.

El desarrollo científico y tecnológico es relevante a nivel estratégico y es por ello que tanto las Fuerzas Armadas en general como el Ejército en particular.

El Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF) realizó aportes a través de Proyectos de Investigación Científico Tecnológicos Orientados (PICTO) para la realización durante 6 años del proyecto recientemente finalizado sobre "REDES PRIVADAS COMUNITARIAS", -dentro del cual se llevó adelante gran parte de esta investigación-, cuyos resultados parciales o finales han sido presentados en varios CACIC para su difusión a la comunidad científica.

4. Introducción.

4.1. Generalidades

La Criptografía dejó de pertenecer a la esfera de los secretos militares y diplomáticos y se ha volcado al ámbito civil en general. No han dejado de crecer sus aplicaciones: ingreso a redes sociales, servidores de mails, home banking, compras online, redes WI-FI, almacenamiento de información confidencial o sensible, etc.

Todo lo que implique intercambio de información entre dos equipos informáticos, debería estar al resguardo de ojos indiscretos, sobre todo los intercambios inalámbricos.

Los sistemas se encargan de entablar todos los servicios requeridos por los estándares y protocolos de manera automática: tickets de ingreso a sistemas, intercambios de claves, autenticación de usuarios y equipos, inicio de sesión, y muchos otros son controlados en forma automática. Los usuarios por lo general no cambian las opciones por defecto: por ejemplo las personas que instalan sus routers con servicios inalámbricos sin autenticación de usuarios ni protocolos de encripción.

¿Cómo comprobar si estos procesos y servicios contienen errores que pueden alterar la seguridad de lo que pretenden proteger?

4.2. Código abierto no necesariamente aumenta la Seguridad de los Sistemas de Información.

El uso de Software Abierto (open source) en el ámbito de la Seguridad de la Información, es mucho más atractiva que la de software cerrado dado que se puede "ver" lo que el sistema hace y cómo lo hace. La ausencia de "puertas traseras" o claves no documentadas u otras técnicas como la inoculación de código virulento. Aunque estén abiertos sus códigos no significa que se encuentren libres de errores.

No todos los usuarios pueden leer el código e interpretar su significado y de esa manera corroborar el correcto funcionamiento del servicio. Luciano Bello ha descubierto un error en

OpenSSL[2]. Que fue enmendado 20 meses después que la versión defectuosa. [02].

Según la Ley de Linus "dado un número muy grande de ojos, los errores se convierten en evidentes" [03].

Está claro que al tiempo que aumenta la sofisticación y complejidad de los códigos, la cantidad de ojos disponibles para evidenciar los errores decrece. Es por ello que automatizar el análisis del funcionamiento del software en busca de anomalías se hace cada vez más necesario.

4.3. Planteamiento del problema.

Un oráculo tipo Open-SSL que ofreciera protección criptográfica (en sus diferentes formas antes mencionadas: claves de sesión, autenticación de equipos, etc.) es vulnerable cuando:

1) Oráculo genera un número relativamente pequeño de primos[3].

2) Oráculo está diseñado con una directiva errónea para la selección de los primos o la generación de los módulos.

En ambos casos Oráculo se distancia del comportamiento equiprobable, dentro de ciertos parámetros asumidos, a un comportamiento sesgado. Susceptible de ser estudiado por un atacante y vulnerar así el sistema.

3) Un atacante tiene forma de construir *un subconjunto P´de Primos con los que trabaja* tal que el conocimiento de P´ le permita vulnerar la factorización de RSA para una cantidad significativa de módulos.

Vemos que se deben hallar los factores primos del esquema RSA, proceso complejo pues esa es la fortaleza que permite a esta criptosistema proteger la información: la complejidad en la factorización de un número enorme.

5. Resultados y Objetivos.

5.1. Resultados intermedios de la línea de Investigación.

En el año 2008 nuestra investigación se oriento a la elaboración una herramienta informática que permite hallar los primos que componen un módulo RSA con la información que aportan su clave pública y su clave privada, resultado presentado y publicado en CACIC 2008 [04] realizado en La Rioja, Argentina.

Las posteriores pruebas de codificación e implementación demostraron que este procedimiento corría muy veloz presentándose estos resultados en CUBA [05].

Seguimos aún estudiando el comportamiento de este algoritmo y lo comparamos con el procedimiento que aborda el mismo problema y lo resuelve, existente en la bibliografía tradicional para la enseñanza de la criptografía [06].

Los análisis indicaron que la complejidad computacional de nuestro algoritmo era del orden O (log n) mientras que el de la bibliografía de referencia tenía un orden O ($\log^3$ n). Resultado presentado en Chile y que nos sorprendió gratamente [07].

Hallada la herramienta matemática que permitiría detectar anomalías (en caso que

[2] Una mala inicialización de una variable provocó una predictibilidad en el generador de números, abriendo una vulnerabilidad inimaginable.

[3] Vulnerabilidad de OpenSSL de Devian descubierta por L. Bello.

las hubiere) el resultado publicado en CACIC 2011 [08] en La Plata.

Con el abordaje probabilístico del problema presentado en 41 JAIIO [09] y por último el diseño final de la herramienta probabilística y estadística presentado en CACIC 2012 [10] en Bahia Blanca se terminó de dar forma definitiva al sustento teórico.

Simultáneamente al avance matemático se ensamblaron y codificaron todas las herramientas matemáticas antes mencionadas, en una plataforma de software programado en C++.

5.2. Etapa actual

Se llevaron adelante pruebas en las que, por lotes, se le solicita a OPEN-SSL la entrega de módulos RSA a efectos de buscar colisiones de primos. Es decir, se analizan los factores primos de cada uno de ellos y se los compara con los de los otros módulos del lote en busca de factores repetidos.

En primer lugar estamos buscando las causas por las cuales el proceso informático agota la memoria de la computadora y el sistema operativo interrumpe la ejecución del programa. Aún no encontramos la causa y es por ello que la exploración sólo se ha realizado en lotes de 1000 módulos cada uno y para 64 bits de tamaño de los mismos.

Pese a esta dificultad se han podido evaluar 9.000.000 de módulos agrupados en 9000 lotes de 1000 cada uno. Los resultados obtenidos no responden a las predicciones teóricas.
Actualmente estamos revisando todo en búsqueda de respuestas a las inconsistencias halladas. Habría 2 posibilidades a tener en cuenta:

a) la detección de errores en la implementación de la herramienta matemática y su depuración.

b) la detección de una Anomalía en la distribución de factores primos del Oráculo, justo lo que la herramienta busca hallar.

5.3. Etapa final.

Una vez que aseguremos la ausencia de errores, en caso que los hubiera, manipularemos a Oráculo para que presente vulnerabilidades. Correremos el programa y de manera aleatoria repetiremos el experimento con oráculos vulnerables como así también sin manipular.

Se espera que el programa sea capaz de detectar los oráculos vulnerables y alertar acerca de ellos. Asimismo informar también sobre los oráculos seguros.

El experimento se repetirá gran cantidad de veces, registrando los falsos positivos como los negativos si los hubiere. De esa forma estudiar la eficacia y eficiencia de esta herramienta.

Cabe aclarar que dada la magnitud en tamaño y la enorme cantidad de valores con las que trabajan estos oráculos, el abordaje es probabilístico-estadístico.

Publicaremos estos resultados en los próximos meses. Es nuestra intención poder hacerlo en el CACIC 2013.

6. Formación de Recursos Humanos.

Algunos algoritmos fueron codificados y probados en el contexto de la Cátedra de Computación I a cargo del Ing. Mg. Alejandro Repetto, que posee nuestra facultad en la carrera de Ingeniería Informática,

Si bien aún el laboratorio no posee ningún becario doctorando o post-doctorando, el resto de los integrantes del equipo son investigadores categorizados en el programa de Incentivos del Ministerio de Ciencia, Tecnología e Innovación Productiva y en el Sistema Científico Tecnológico de la Defensa en el régimen de Régimen para el personal de Investigación y Desarrollo de las Fuerzas Armadas (RPIDFA).

7. Referencias y Bibliografía

[01] Young A and Yung M. An Elliptic Curve Asymmetric Backdoor in OpenSSL RSA Key Generation. Chapter 10. Cryptovirology. 2006. http://www.cryptovirology.com.

[02] Bello L, Bertacchini M. "*Generador de Números Pseudo-Aleatorios Predecible en Debian"*. III Encuentro Internacional de Seguridad Informática. Manizales, Colombia. Octubre 2009.

[03] Glass, Robert "*Facts and Fallacies of Software Engineering"*. Addison-Wesley Professional, 2003.

[04] Cipriano, M. "*Factorización de N: recuperación de factores primos a partir de las claves pública y privada*." Anales del XIV Congreso Argentino de Ciencias de la Computación CACIC 2008. Chilecito, La Rioja, Octubre 2008.

[05] Castro Lechtaler, C; Cipriano, M; Benaben A; Quiroga, P. "*Study on the effectiveness and efficiency of an algorithm to factorize N given e and d"* Anales del IX Seminario Iberoamericano en Seguridad de las Tecnologías de la Información, La Habana, CUBA. 2009.

[06] Menezes, A; van Oorschot, P and Vanstone, S. *Handbook of Applied Cryptography*. CRC Press. 5th Edition, 2001.

[07] Benaben, A; Castro Lechtaler, A; Cipriano, M; Foti, A. "*Development, testing and performance evaluation of factoring algorithms whit additional information*" XXVIII Conferencia Internacional de la Sociedad Chilena de Computación. Santiago de Chile

[08] Castro Lechtaler, A; Cipriano, M. "*Detección de anomalías en Oráculos tipo OpenSSL por medio del análisis de probabilidades*" Anales del XVII Congreso Argentino de Ciencias de la Computación CACIC 2011. La Plata, Buenos Aires, Octubre 2011.

[09] Castro Lechtaler, Antonio, Cipriano Marcelo; Malvacio Eduardo; Cañón, Sebastián; *Procedure for the Detection of Anomalies in Public Key Infrastruture (RSA Systems).* Anales del XIII Simposio Argentino de Tecnología, 41 Jornadas Argentinas de Informática e Investigación Operativa JAIIO – SADIO. La Plata, Buenos Aires, Agosto 2012.

[10] Castro Lechtaler, Antonio; Cipriano, Marcelo; Malvacio, Eduardo. *Experimental detection of anomalies in public key infrastructure* . Anales del XVIII Congreso Argentino de Ciencias de la Computación CACIC 2012. Bahía Blanca, Buenos Aires, Octubre 2011.