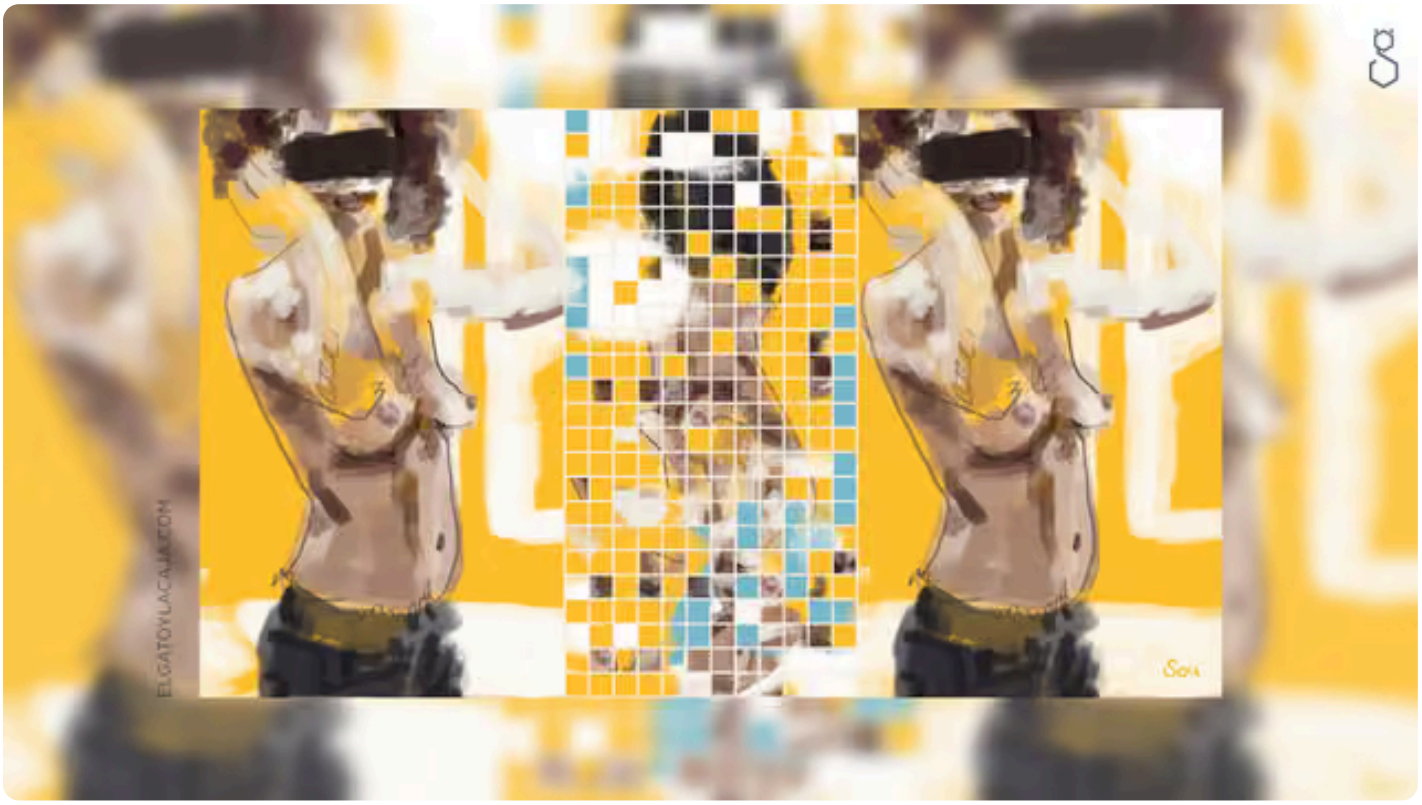

05/10/2015

Tu secreto

TXT AGUSTÍN MARTINEZ SUÑÉ IMG SONIA BASCH

¿Cómo se guarda un secreto en una computadora? ¿Podemos cifrar la información para que nadie la vea?

Disfruto volver a asombrarme con las tecnologías que usamos cotidianamente. Hace apenas un par de días, mientras investigaba la posibilidad de mejorar mi desempeño en una actividad mediante la repetición exhaustiva (o sea, mientras trataba de pasar el nivel 76 del Candy), vi mi pantalla invadida por la publicidad de una aplicación para pagar servicios e impuestos online. Me quedé mirandola por un rato. Decidí momentáneamente sacarme los

anteojos del Siglo XXI y me puse a pensar lo increíble que es que yo pueda, desde el sillón de mi casa, en calzones y pantuflas, pagar el gas, la luz y demás garrones con sólo apretar un botón en la palma de mi mano.

Me pregunté entonces dónde estaba la plata, los billetes, el oro, la sal, ALGO que hiciera alusión a un **intercambio de** dinero, pero no había billetes por ningún lado, nada. Sólo yo en calzones y pantuflas. Bueno, yo y **esa gigantesca red de computadoras interconectadas que llamamos Internet**.

Ponele que agarrás un par de cables de red y conectás tu compu con la de dos vecinos para jugar los tres juntos al Counter Strike (todo con mucha selección sexual positiva, ¿no? No). Bueno, Internet es como eso pero a gran escala, interconectando computadoras de (casi) todo el mundo, y con ciertos mecanismos, o *protocolos,* que sirven para **identificar unívocamente cada una de esas computadoras y poder transmitir datos entre cualquier par de ellas.**

El tema es que a veces las distancias son mayores que a la casa del vecino, pero la necesidad de jugar al Counter es grande (y la de ver porno, ni hablar). Tanto que pusimos **cables submarinos que atraviesan el océano** y que nos conectan con, por ejemplo, Europa. Así es como puedo comunicarme con mis tíos de España y que me muestren en mega HD la paella que se están mandando, sin que los mejillones tengan que rebotar en un satélite.

Pero hay un problema: **como esto es una red**, para llegar a la casa de mi tío en España, los *paquetes de datos* que yo envío tienen que pasar por un montón de otras computadoras en el camino, que llamamos *nodos intermedios*. Ahora, ¿qué me asegura que no va a haber un tipo en Portugal interceptando los datos que envío y cagándose de risa con la transmisión en la que me estoy morfando una tira de asado con las manos para hacerle competencia a la paella de mis tíos? Y sí, uso este ejemplo porque nadie quiere imaginarse un intercambio con Ingrid, la hermosa chica alemana que conocí en un congreso, PORQUE ES MI INTIMIDAD Y LA VAN A RESPETAR. Pero, como si con el asunto de la privacidad cotidiana no fuese suficiente, **la cuestión se vuelve crítica si lo que estoy enviando son los datos de mi tarjeta de crédito o datos para hacer una transferencia bancaria.**

Uno pensaría que no pasa una, que es todo re seguro, re lindo y re sólido, pero la verdad es que **absolutamente nada me garantiza que no va a haber alguien interceptando la información que envío a través de Internet**.

Todo mal entonces, ya fue, **volvamos al trueque.** El tema es que pagar la factura del celular con vacas ya no es muy práctico, así que vamos a tener que probar otra cosa. ¿Y si mandamos el mensaje de manera tal que sólo el receptor final pueda entenderlo? En otras palabras, ¿qué tal si **ciframos el mensaje antes de enviarlo**? O sea, en lugar de enviarlo así como está (lo que llamamos *texto plano*), lo transformamos en algo inentendible, en *texto cifrado*, algo que sólo pueda entenderse si se vuelve a transformar a texto plano. De esta manera, **la seguridad del método reside en que sólo el receptor final pueda *desencriptar* el mensaje.**

Este es el punto de partida de lo que se conoce como *criptografía*, una disciplina muy poderosa que desde la antigüedad genera herramientas que permiten **enviar información de manera segura.** Y no me refiero sólo a que no se filtren tus fotitos de WhatsApp en donde procurás que nunca coincidan cara o tatuaje con regiones donde no da el Sol. Hablo de cosas grosas, como enviar indicaciones militares en la Segunda Guerra Mundial. Bueno, y fotos de WA más seguras también, tranquilo. Claro que, como muchos avances científicos y fotos de WA, su desarrollo no siempre fue guiado por los intereses más inocentes y altruistas del mundo.

Los procedimientos para hacer estas transformaciones, o mejor dicho ***los algoritmos de encriptación,* se forman a partir de una *clave secreta y* consisten en cuentas matemáticas que toman como entrada el mensaje en texto plano y lo transforman en un mensaje encriptado**. Por su parte, los *algoritmos de desencriptación* hacen la tarea inversa, de manera tal que sólo quien posea la clave secreta puede aplicar el algoritmo de forma correcta para recuperar el mensaje.

Esto es todavía más claro visto en una situación concreta, así que resucitemos la red que armamos con esos dos vecinos para jugar al Counter Strike. Bueno, resulta que el pibe del 2do B, Juan, siempre te pareció muy fachero y ahora que se conocen mejor lo querés invitar a salir.

Obvio que no te animás a decírselo en la cara, así que vas a conquistarlo con mensajitos por la red (porque todos sabemos que así funcionan las cosas ¿no?); pero no querés que Laura, la otra vecina que también está en la red, se entere de nada. Primero, por privacidad y, segundo, porque está bastante buena y como que las comparaciones son fatales. El tema es que antes de poder entablar cualquiera de esas conversaciones subiditas de tono que tanto te gustan, **es necesario que Juan tenga la clave para desencriptar tus mensajes**. Pero no se la podés mandar por la red, ¡porque Laura podría verla también! O sea que **el intercambio de claves de manera segura es un problema en sí mismo.**

Una opción es que vayas a tocarle la puerta y le des la clave en persona, es decir, que la **transmitas a través de un *canal seguro.*** En este caso usarían una misma clave secreta para encriptar y desencriptar todos los mensajes, lo que se llama *criptografía de clave simétrica.* Pero no siempre se cuenta con un canal seguro para intercambiar claves, y la posta es que ni a palos te animás a hablarle en persona. Acá es muy útil usar lo que se llama *criptografía de clave pública,* en la que cada participante tiene dos claves: una clave pública, que sirve para que los demás encripten mensajes, y **una clave privada, para desencriptar esos mensajes.** En este caso, si Laura consigue la clave pública no es un problema, ya que **sólo vos tenés la clave privada** que sirve para descifrar los mensajes que fueron encriptados con tu clave pública. **Tu clave pública y tu clave privada están relacionadas matemáticamente y es por esto que una sirve para desencriptar los mensajes que se encriptan con la otra.** Sólo queda que Juan también genere un par de claves y te mande su clave pública para que vos puedas mandarle mensajes cifrados a él.

Con esta idea en la cabeza, **en el '77 tres tipos inventaron el primer método para llevar esto a una implementación práctica.** Hoy lo conocemos como RSA (por las iniciales de sus apellidos), y es uno de los sistemas criptográficos más utilizados en el mundo.

Ahora, estamos hablando de algo que pasó hace casi cincuenta años. **¿Cómo puede ser que hasta hoy, con computadoras miles de veces más poderosas, nadie haya podido vulnerar ese sistema?**

La seguridad de un sistema de clave pública reside en que, **si bien la clave pública y la clave privada están relacionadas matemáticamente, no debe ser posible calcular la clave privada a partir de la clave pública (o, por lo menos, no dentro de un umbral de tiempo lógico).** En el caso de RSA, las claves consisten de números muy grandes, números gigantescos. Y, por la relación matemática que propone RSA, la forma de calcular una clave privada a partir de su clave pública es tomando una parte de esta y **obteniendo su descomposición en factores primos.**

Sí sí, lo mismo que esas tablitas que nos hacían hacer en la escuela: 10 = 2 x 5; 6 = 3 x 2... Suena re fácil, ¿no?

No. Resulta que hasta el momento **no existe ningún algoritmo que resuelva la factorización de números tan grandes en un tiempo razonable.** Los métodos actuales tardarían cientos de miles de años en encontrar la solución. O sea que de la única persona que realmente deberías cuidarte es de Mirtha.

Lo que sí se puede hacer en tiempo razonable es verificar si una solución es válida. Por ejemplo, si me dicen que la factorización de 123 es 3 x 41, basta con multiplicarlos y ver que el resultado es correcto. Como este, **hay bocha de problemas en los que, si bien se puede verificar fácilmente que una solución dada es correcta, lo que no se puede es calcular la solución de manera eficiente.** No importa si usás el procesador último modelo de Intel con 700 núcleos, enanos, gaseosa y papas grandes. No alcanza con hacer procesadores más rápidos, porque *es un problema algorítmico*. **La dificultad está en que el tiempo que se tarda en calcular la solución de estos problemas crece de manera gigantesca (exponencial) a medida que se tienen datos de entrada cada vez más grandes.**

Y lo más flashero de todo es que, en realidad, **no se sabe si es que no existen formas eficientes de resolver estos problemas o si es que todavía a nadie se le ocurrió el algoritmo que los resuelva en un tiempo razonable**. De hecho, esta es una de las preguntas abiertas más importantes de las *ciencias de la computación*, conocida como '*P vs NP*'. Tan importante es que, si la resolvés, te dan un premio de un millón de dólares. Y un

abrazo. Y acceso a todos los secretos de Estado de la historia del mundo. Y el celular de Megan Fox.

Así que atenti, porque esto de factorizar números no es joda. Si te ponés las pilas, en una de esas te hacés millonario y tirás abajo todos los sistemas de seguridad del planeta. Todo por intentar comerte al vecino. O a Megan Fox. O a los dos.

Referencias

https://www.iscpc.org/cable-data/

Peterson, Davie: Computer Networks: A Systems Approach

http://en.wikipedia.org/wiki/Enigma_machine

Rivest, R.; Shamir, A.; Adleman, L. (February 1978). “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”.

http://download.intel.com/pressroom/kits/IntelProcessorHistory.pdf

Garey M.R. and Johnson D.S., “Computers and intractability: a guide to the theory of NP-Completeness”.
