TUGAS II
RESUME
“KEAMANAN SISTEM KOMPUTER”

DISUSUN OLEH :

NAMA : ANDI AMANDA ANDI TALLAGU

NIM : F55122034

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TADULAKO
PALU

2023

I. CONFIDENTALLY WITH SYMETRIX ENCRIPTION

Elemen paling penting dalam banyak layanan dan aplikasi keaman komputer adalah penggunaan algoritma kriptografi, dimana pembahasan ini akan memberikan gabmabaran umum tentang berbagai jenis algoritma, dimulai dengan snkripsi simetris dimana hal ini digunakan dalam berbagai macam konteks terutama dalam mengamankan sebuah kerahasiaan.

Teknik yang sering digunakan adalah untuk memberikan kerahasiaan untuk data yang ditransmisikan atau disimpan adalah enkripsi simetris kemudian diikuti dengan tnjauan dua algoritma yang paling penting yaotu Data Encryption dan Advanced Encryption Standard yang merupakan algoritma enskripsi blok.

A. Symetric Encription

Enkripsi simetris juga disebut sebagai enkripsi konvensional atau enskripsi kunci tunggal yang merupakan satu-satunya jenis enkripsi yang digunakan sebelum diperkenalkannya enkripsi kunci kepada publik pada kahir tahun 1970-an, dimana terdapat banyak sekali individu dan kelompok dari Julius Caesar, pasukan U-boat Jerman hingga pengguna diplomatik, militer dan komersial saat ini telah menggunakan enkirpsi simetris untuk komunikasi rahasia, selain itu enkripsi ini juga memiliki lima bahan yaitu :

1. Paintext : ini adalah pesan atau data asli yang dimasukkan ke dalam algoritma sebagai masukkan
2. Algoritma Enkripsi : dimana algoritma ini melakukan berbagai subsitusi dan transformasi pada plaintext
3. Kunci Rahasia : kunci rahasia ini merupakan sebuah masukkan untuk algoritma enkripsi , subsitusi dan transformasi yang tepat yang dilakukan oleh algoritme bergantung pada kunci
4. Chriptex : ini adalah pesan yang diacak yang dihasilkan sebagai output, dimana beberapa hal tergantung pada plaintext dan kunci rahasia, dalam mmebrikan pesan perbedaan keduanya akan menghasilkan dua ciphertext yang berbeda
5. Algoritman deskripsi : pada dasarnya algoritma enkripsi akan terjalankan secara terbalik, algoritma ini mengambil ciphertext dan kunci rahasia dan menghasilkan plaintext asli

Ada dau sayarat yang diperlukan dalam penggunaan enkripsi simetris yang aman yaitu :

- Mereka butuh Algoritma enkripsi yang kuat, setidaknya , mereka menginginkan algoritma tersebut sedemikian rupa hingga lawan yang mengetahui algoritma tersebut dan memiliki akses ke satu atau beberapa ciphertext tidak akan dapat menguraikan ciphertext atau mengetahui kuncinya biasanya persyaratan dinyatakan dalam bentuk yangg lebih kuat : lawan tidak akan dapat mendeksripsikan ciphertext atau menemukan kuncinya walaupun ia memiliki sejumlah ciphertext bersama dengan text biasa yang menghasilkan setiap ciphertext
- Pengirim dan penerima harus mendapatkan salinan kunci rahasia dengan cara yang aman dan harus menjaga kunci tersebut tetap aman, jika seseorang dapat menemukan kunci tersebut dan mengetahui algoritma maka semua komunikasi yang menggunakan kunci ini dapat dibaca

Terdapat dua hal umum yang digunakan penyerang untuk pendekatan dalam menyerang skema enkripsi simetris yaitu **serangan pertama dikenal sebagai kriptanalisis** dimana serangan ini bergantung pada sifat dari algoritma yang juga ditambah dengan pengetahuan tentang karakter umum dari plaintext atau bahkan beberapa contoh pasangan palintext-ciphertext. Jenis serangan ini mengeksploitasi karakteristik algoritma untuk mencoba menyimpulkan kunci yang dgunakan, jika serangan tersebut berhasil menyimpulkan kunci maka efeknya akan menjadi masalah besar dimaan semua pesan akan datang dan sydah dienskripsi dengan kunci yang akan dibobol.

Srangan kedua dikenal sebagai serangan brute-force, metode serangan ini mencoba semua kunci yang mungkin terdapat pada sepotong ciphertext sampai terjemahan yang dapat dimengerti ke dalam plaintext, terdapat rata ratasetengah dari semua kunci yang mungkin harus dicoba untuk mencapai keberhasilan dengan kata laun, jika terdapat x kunci yang berbeda maja rata-rata penyerang akan menemukan kunci yang sebenarnya setelah x/2 percobaan. Penting untuk dicatat bahwa ada lebih banyak hal pada serangan brute-force daripada hanya menjalankan semua

kunci yang mungkin terjadi, kecuali jika plaintext yang diketahui disediakan analis harus dapat mengenali palintext sebagai plain text

B. Messege Authentication And Hash Functions

Enkripsi melindungi dari serangan pasif(penyadapan). Dimana terdapat persyarata berbeda yaitu melindungi dari serangan aktif (pemalsuan data dan transaksi), perlindungan terhadap serangan semacam itu dikenal sebagai otentikasi pesan atau data

Dimana sebuah pesan, dokumen, dan kumpulan data data lainnya dikatakan autentik jika pesan, file, dokumen atau kumpulan data tersebut asli dan berasal dari sumber yang seharusnya, otentikasi pesan atau data adalah prosedur yang memungkinkan pihak pihak yang berkomunikasi untuk memverifikasi bahwa isi pesan belum diubah dan bahwa sumbernya asli.

1. Authentication Using Symetric Encryption

Tampaknya akan terlihat lebih memungkinkan untuk menmapilkan autentikasi dengan menggunakan enskripsi simetris, jika kita mengasumsi kan bawa hanya pengiri dan penerima yan berbagau kunci (sebagimana mestinya), maka hanya pengirim asli yang dapat mengenali pesan yang valid. Selain itu jika pesa menyertakan kode pendeteksi kesalahan nomor urut, penerima diyakinkan bahwa tidak ada perubahan yang telah dilakukan dan urutan tersebut benar dan jika pesan juga menyertakan stampel waktu, yang biasanya diharapkan untuk transit jaringan, namun pada kenyataanya enkripsi simetris saja bukanlah alat yang cocok untuk autentikasi data, sebagai contoh sederhana pada metode enkripsi ECB, jika berhasil dideskripsi. Akan tetapi pengurutan ulang tersebut dapat mengubah arti dari urutan data keseluruhan meskipun nomor urut dapat digunakan pada beberapa tingkat(misalnya, setiap paket IP), biasanya tidak ada nomor urut yang terpisah yang akan diasosiasikan dengan setiap blok b-bit dari plaintext dengandemikian pengurutan ulang blok adalah sebuah ancaman

a. Otentikasi Pesan tanpa Enkripsi Pesan

Pada bagian ini, kami memeriksa beberapa pendekatan untuk autentikasi pesan yang tidak bergantung pada enkripsi pesan, pada semua pendekatan ini, sebuah tag

autentikasi dibuat dan ditambahkan pada setiap pesan untuk transmisi. Pesan itu sendiri tidak dienkripsi dan dapat dibaca ditempat tujuan tanpa bergantung pada fungsi autentikasi, karena pendekatan yang dibahas pada bagian ini tidak mengenripsi pesan, kerhasiaan pesan tidak disediakan seperti yang telah disebutkan, enkripsi pesan dengan sendirimya tidak menyediakan bentuk autentikasi yang aman, akan tetapi dimungkinkan untuk menggabungkan autentikasi den kerhasiaan dalam sebuah algoritma tunggal dengan mengenkripsi pesan ditambah dengan tag autentikasi [DAVI189] menyarankan tiga situasi di mana autentikasi pesan tanpa kerahasiaan lebih disukai:

1. Ada sejumlah aplikasi di mana pesan yang sama disiarkan ke sejuml,ah tujiam terdapat dua contoh yaitu pemberitahuan kepada pengguna bahwa jaringan sekarang tidak tersedia, dan sinyal sinyal alaram dipusat kendali, akan lebih murah dan lebih dapat diandalkan jika hanya ada satu tujuan yang bertanggung jawab untuk memonitori keaslian
2. Skenario lain yang mungki terjadi adalah pertukaran dimana salah satu pihak memiliki beban yang berat dan tidak memiliki waktu untuk mendeskripsikan semua pesan yang masuk, autentikasi dapat dilakukan secara selektif dengan pesan dipilih secara acal untuk diperiksa
3. Otentikasi program komputer dalam bentuk plaintext adalah layanan yang menarik, program komputer dapat dieksekusi tanpa harus mendeskripsi setiap saat yang akan memboroskam sumber daya prosesor akan tetapi jika sebuah pesan tag otentikasi dilampirkan pada program, maka dapat diperiksa kapanpun diperlukan jaminan integritas program

2. Fungsi Hash yang Aman

   Fungsi Hash satu arah atau fungsi hash aamn tidak hanya penting dakam otentikasi pesan tetapi juga dalam tanda tangan digital. Pada bagian ini kita mulai dengan pesan diskusi tentang persyaratan untuk fungsi hash yang aan dan kemduia kita akan mendiskusikan algoritma yang spesifik

   a. Persyaratan FungsI Hash

   Tujuan fungdi ini adalah untuk menghasilkan “sidik jari” dari sebuah file, pesan, atau blok data lainnya.

Agar dapat digunakan untuk autentikasi pesan fungsi hash harus memiliki sifat-sifat berikut

- H dapat diterapkan ke blok data dengan ukuran berapa pun
- H menghasilkan output dengan panjang
- H(x) relatif mudah dihitung untuk setiap x yang diberikan, membuat implementasi perangkat keras dan perangkat lunak menjadi praktis
- Untuk setiap kode h yang diberikan secara komputasi tidak mumgkin untuk menemukan x sedemikan rupa sehingga H(x) h fungsi hash dengan properti ini disebut sebagai satu arah atau tahan terhadap gambar
- Untuk setiap blok x yang diberikan secara komputasi tidak mungkin menemukan y!=x dengan H(y) H(x). sebuah fungsi hash dengan properti ini disebut sebagai preim-kedua
- Secara komputasi tidak mungkin untuk menemukan pasangan(x,y) yang sedemikian rupa sehingga H(x) H(y). fungsi hash dengan properti ini disebut sebagai tahanan tabrakan, ini terkadang disebut sebagai kuat terhadap tabrakan

b. Algoritma Fungsi Hash Aman

Dalam beberapa tahun terkahir fungsi hasj paling banyak digunakan adalah secure hash Algorithm (SHA). SHA di kembangkan oleh national Insititue of standards and Technology, dan diterbitkan sebagai standar pemrosesan informasi federal(FIPS 180)

3. Aplikasi Lain dari Fungsi Hash

- Kata sandi, diaman hass kata sandi disimpan oleh sistem operasi dan bukan kata sandi itu sendiri dengan demikian kata sandi yang sebenarnya tidak dapat diambil oleh peretas yang mendaptkan akses ke file kata sandi, secara sederhana ketika pengguna memasukkan kata sandi hash dari kata sandi

tersebut dibandingkan dengan nilai hash yang tersimpan untuk verfikasi aplikasi ini membutuhkan ketahanan preimage dan mungkin ketahanan preimage kedua

- Deteksi penyusupan, menuimpan nilai hash dalam sebuah file H(f), untuk setiap file pada sebuah sistem dan mengamankan nilai hash tersebut, misalnya, pada CD-R yang dijaga keamanannya, seseorang dapat menentukan apakah penyusup perlu mengunah F tanpa mengunah H(F), aplikasi ini membutuhkan ketahanan preimage kedua yang kemah

C. Public Key Encryption

1. Public Key Encryption Structure

Pertama kali diusulkan secara publik oleh Diffie dan Helman pada thun 1976, algoritma kunci publik didasarkan pada fungsi matetmatika dan buka pada operasi sederhana pada pola bit, seperti yang digunakan pada enkripsi simetris, yang perlu diektahui adalah kriptografi kunci publik bersifat asimetris, yang mana melibatkan penggunaan dua kunci yang terpisah berbeda dengan kunci publik ang berisfat simetris yang hanya menggunakan satu kunci, penggunaan dua kunci memiliki kosekuensi yang ditemukan di bidang kerh=ahaisaan, distribusi kunci, dan autentikasi.

Kesalahpahaman yang terjadi ada enkripsi kunci publik adalah enkripsi kunci publik lebih aman dari analisi kriptanalisis dari enkripsi simetris

Dimana keamanan setiap skema enkripsi bergantung pada panjang kunci danpekerjaan komputasi yang dibutuhkan untuk memecahkan sebuah sandi, kesalhpaham yang kedua yaitu bahwa enkripsi kunci publik memiliki tujuan umum yang mebuat enkripsi simetris menjadi usang, sebaliknya karena overhead komputasi dari skema kemungkinan enskripsi ini akan ditinggalkan

Ada beberapa sekam yang dimiliki selama 6 bahan yaitu :

- plainText, adalah pesan atau data yang dapat dibaa yang dimasukkan ke dalam algoritma sebagai masukan

- algoritma enkripsi, melakukan berbagai transformasi pada plaintext
- kunci publik dan pribadi, adalah sepasang kunci yang telah dipilih sehingga jika salah satu digunakan untuk enkripsi yang lain digunakan untuk deskripsi, transformasi yang tepat dilakukan oleh algorita enkripsi bergantung pada kunci publik atau privat yang disediakan sebagai masukkan
- ciphertext, pesan yang diacak yang dihasila sebagai output hal ini tergantung pad aplaintext dan kuncinya
- Algoritma deskripsi, menerima ciphertext dan kunci yang coock dan menghasilkan plaintext yang asli

Seperti namanya kunci publik dari pasangan ini dipublikasikan untui dketahui oleh orang lain, sedangkan kunci private hanya diketahui ileh pemiliknya saja. Algoritma Kriptografi kunci publik untuk tujuan umum bergantung pada satu kunci untuk enkripsi dan kunci yang bebrbda tetapi terkait untuk deskripsi

Dimana terdapat langkah langkah penting sebagai berikut:

a. Setiap pengguna menghasilkan sepasang kunci yang akan digunakan untuk enkripsi dan deskripsi pesan
b. Setiap pengguna menmempatkan salah satu dari dua kunci tersebut ke dalam daftar publik atau file lain yang dapat diakses, dimana kunci pendamping disimpan secara pribadi yang artinya setiap pengguna menimpan sebuah koleksi kunci publik yang didaptkan dar pengguna lain
c. Jika Bob ingin mdngirim pesan pribadi pas Alie maka, Bob mengenkripsi pesan tersebut menggunakankunci publik Alice
d. Ketika Alice menerima pesan tersebut dia mendeskripsi pesan tersebut menggunakan kunci pribadinya, tidka ad apenerima lain yang dapat mendeskripsi pesan tersebut karena hanya Alice yang mengetahui

D. Algoritma Enkripsi Asimetris

1. **RSA** Salah satu skema kunci publik pertama dikembangkan pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman di MIT dan pertama kali dipublikasikan pada tahun 1978 [RIVE78]. Skema RSA sejak saat itu menjadi yang tertinggi sebagai pendekatan yang paling banyak diterima dan diimplementasikan untuk enkripsi kunci publik. RSA adalah sebuah blok cipher dimana plaintext dan ciphertext adalah bilangan bulat antara 0 dan n - 1 untuk beberapa n.
2. **Perjanjian Kunci**, Diffe-Hellman Algoritma kunci publik yang pertama kali dipublikaiskan muncul dalam makalah penting oleh diffie dan hellman. Tujuan dari algoritma ini adalah untuk memungkinkan dua pengguna untuk mencapai kesepakatan dengan aman tentang rahasia bersama yang dapat digunakan sebagai kunci rahasia untuk enkripsi simetris pesan selanjutnya. Algoritme itu sendiri terbatas pada pertukaran kunci
3. **Standar Tanda Tangan Digital**, National Institue Of Standards and Technology (NIST) telah menerbitkan standar Tanda Tangan Digital (DSS) dimana DSS ini menggunakan SHA-1 dan menyajikan teknik tanda tangan digital baru, yaitu Algoritma Tanda Tngan Digital. DSS menggunakan sebuah algoritma yang didesain untuk menyediakan fungsi tanda tangan digital saja. Tidak seperti RSA, DSS tidak dapat digunakan untuk enkripsi atau pertukaran kunci.

E. Digital Signature and Key MANAGEMENT

Algoritma ku ci publik digunakan dalam berbagai jenis aplikasi, secara garis besae aplikasi ini terbagi ke dalam dua kategori : tanda tangan dgital dan berbagai teknik yang berhubungan dengan manajemen dan distribusi kunci dalam hal ini terdapat 3 aspek yang berbeda dalam enkripsi kunci publik yaitu :

• Distribusi kunci publik yang aman

• Penggunaan enkripsi kunci publik untuk mendistribusikan kunci rahasia

• Penggunaan enkripsi kunci publik untuk membuat kunci sementara untuk enkripsi pesan

Berikut gambaran singkat mengenai tanda tangan digital dan berbagai jenis manajemen dan distribusi kunci

1. **Tanda Tangan Digital**

   Misalkan Bob ingin mengirimkan sebuah pesan ke Alice, meskipuntidak penting akan tetapi pesan itu bersifat rahasia, Bob ingin meyakinkan Alice bahwa pesan ini bersasal dari Bob, dari tujuan ini Bob menggunakan keamanan Fungsi Hash seperti SHA-512 untuk menghasilkan nilai Hash pada pesan tersebut, ketika Alive menerima pesan dan tanda tangan tersebut hal pertama yang ia lakukan adalah :

   - Menghitung nilai Hash untuk pesan tersebut
   - mendekripsi tanda tangan tersebut dengan menggunakan kunci publik Bob
   - membandingkan nilai hash yang telah dihitung dengan nilai hash yang telah didekripsi. Jika kedua nilai hash tersebut cocok, Alice yakin bahwa pesan tersebut pasti ditandatangani. Tidak ada orang lain yang memiliki kunci pribadi Bob dan oleh karena itu tidak ada orang lain yang dapat membuat ciphertext yang dapat didekripsi dengan kunci publik Bob. Selain itu, tidak mungkin untuk mengubah pesan tersebut tanpa akses ke kunci privat Bob

2. **Sertifikat Kunci Publik**

   Secara sepintas, inti dari enkripsi kunci publik adalah bahwa kunci publik bersifat publik. Dengan demikian, jika ada beberapa algoritma kunci publik yang diterima secara luas, seperti RSA, setiap partisipan dapat mengirimkan kunci publiknya kepada partisipan lain atau menyiarkan kunci tersebut kepada komunitas secara luas. Walaupun pendekatan ini mudah digunakan, pendekatan ini memiliki kelemahan utama. Siapapun dapat memalsukan pengumuman publik tersebut. Dengan kata lain, beberapa pengguna dapat berpura-pura menjadi Bob dan mengirimkan kunci publik kepada peserta lain atau menyiarkan kunci publik tersebut. Sampai pada saat Bob menemukan pemalsuan tersebut dan memberitahukan kepada partisipan lain, pemalsu dapat membaca semua pesan terenkripsi yang ditujukan kepada Bob dan dapat

menggunakan kunci yang dipalsukan tersebut untuk otentikasi

3. **Pertukaran Kunci Simetris Menggunakan Enkripsi Kunci Publik**

   Dengan enkripsi simetris, persyaratan mendasar bagi dua pihak untuk berkomunikasi dengan aman adalah mereka berbagi kunci rahasia, diberikan contoh sebagai berikut Misalkan Bob ingin membuat sebuah aplikasi pengiriman pesan yang memungkinkannya untuk bertukar email secara aman dengan siapa saja yang memiliki akses ke Internet atau jaringan lain yang digunakan bersama. Misalkan Bob ingin melakukan ini dengan menggunakan enkripsi simetris. Dengan enkripsi simetris, Bob dan korespondennya, katakanlah, Alice, harus menemukan sebuah cara untuk berbagi sebuah kunci rahasia yang unik yang tidak diketahui oleh orang lain. Bagaimana mereka akan melakukannya? Jika Alice berada di ruangan sebelah dari Bob, Bob dapat membuat sebuah kunci dan menuliskannya di selembar kertas atau menyimpannya di dalam sebuah cakram atau flashdisk dan memberikannya kepada Alice.tetapi ketika Alice berada di belahan dunia lain Bob dapat mengenkripsi kunci ini menggunakan enskripsi simetris dan mengirimkannya melalui email kepada Alice

4. **Amplop Digital**

   Aplikasi lain di mana enkripsi kunci publik digunakan untuk melindungi kunci simetris adalah amplop digital, yang dapat digunakan untuk melindungi pesan tanpa perlu terlebih dahulu mengatur agar pengirim dan penerima memiliki kunci rahasia yang sama. Teknik ini disebut sebagai amplop digital, yang setara dengan amplop tertutup yang berisi surat yang tidak ditandatangani. Pendekatan umum ditunjukkan pada Gambar 2.8. Misalkan Bob ingin mengirimkan sebuah pesan rahasia kepada Alice, tetapi mereka tidak memiliki kunci rahasia yang sama. Bob melakukan hal berikut:

   1. Siapkan pesan.
   2. Hasilkan kunci simetris acak yang hanya akan digunakan sekali ini saja.

3.Enkripsi pesan tersebut menggunakan enkripsi simetris dengan kunci sekali pakai.

4. Enkripsi kunci satu kali menggunakan enkripsi kunci publik dengan kunci publik Alice.

5. Lampirkan kunci satu kali terenkripsi ke pesan terenkripsi dan kirimkan ke Alice. Hanya Alice yang mampu mendekripsi kunci sekali pakai dan oleh karena itu dapat memulihkan pesan asli. Jika Bob mendapatkan kunci publik Alice dengan menggunakan sertifikat kunci publik Alice, maka Bob yakin bahwa kunci tersebut adalah kunci yang valid.

F. Random And Pseudorandom Numbers

1. Penggunaan Angka Acak

   Sejumlah algoritma keamanan jaringan berdasarkan kriptografi menggunakan angka acak. Sebagai contoh,

   • Pembuatan kunci untuk algoritma enkripsi kunci publik RSA dan algoritma kunci publik lainnya.

   • Pembuatan kunci aliran untuk sandi aliran simetris.

   • Pembuatan kunci simetris untuk digunakan sebagai kunci sesi sementara atau dalam membuat amplop digital.

   • Pada beberapa skenario distribusi utama, seperti Kerberos (dijelaskan pada Bab 23), angka acak digunakan untuk jabat tangan untuk mencegah serangan replay.

   • Pembuatan kunci sesi, baik yang dilakukan oleh pusat distribusi kunci atau oleh salah satu prinsipal. Aplikasi ini memunculkan dua persyaratan yang berbeda dan belum tentu kompatibel untuk urutan angka acak: keacakan dan ketidakpastian. Keacakan Secara tradisional, perhatian dalam pembuatan urutan angka yang diduga acak adalah bahwa urutan angka tersebut acak dalam beberapa pengertian statistik yang didefinisikan dengan baik. Dua kriteria berikut ini digunakan untuk memvalidasi bahwa urutan angka adalah acak: • Distribusi yang seragam: Distribusi angka dalam urutan harus seragam; yaitu, frekuensi kemunculan masing-masing angka harus kurang lebih sama.

   • Kemandirian: Tidak ada satu nilai dalam urutan yang dapat disimpulkan dari nilai lainnya

2. Acak Versus Pseudorandom

Aplikasi kriptografi biasanya menggunakan teknik algoritmik untuk menghasilkan nomor acak. Algoritme ini bersifat deterministik dan oleh karena itu menghasilkan urutan angka yang tidak acak secara statistik. Namun, jika algoritmanya bagus, urutan yang dihasilkan akan melewati banyak tes keacakan yang masuk akal. Angka-angka tersebut disebut sebagai angka pseudorandom. Anda mungkin agak tidak nyaman dengan konsep penggunaan angka yang dihasilkan oleh algoritme deterministik seolah-olah itu adalah angka acak. Meskipun ada keberatan filosofis terhadap praktik semacam itu, pada umumnya hal ini berhasil. Artinya, dalam sebagian besar s i t u a s i , bilangan pseudorandom akan bekerja sebaik jika bilangan tersebut diacak untuk penggunaan tertentu. Frasa "sebaik" sayangnya bersifat subjektif, tetapi penggunaan bilangan acak semu diterima secara luas. Prinsip yang sama berlaku dalam aplikasi statistik, di mana seorang ahli statistik mengambil sampel dari suatu populasi dan mengasumsikan bahwa hasilnya akan kurang lebih sama seperti jika seluruh populasi diukur. Generator angka acak sejati (TRNG) menggunakan sumber nondeterministik untuk menghasilkan keacakan. Sebagian besar beroperasi dengan mengukur proses alami yang tidak dapat diprediksi, seperti detektor denyut nadi dari peristiwa radiasi pengion, tabung pelepasan gas, dan kapasitor yang bocor LavaRnd adalah proyek sumber terbuka untuk membuat angka yang benar-benar acak menggunakan kamera yang tidak termenung, kode sumber terbuka, dan perangkat keras yang tidak mahal. Sistem ini menggunakan perangkat chargecoupled device (CCD) jenuh dalam kaleng kedap cahaya sebagai sumber acak untuk menghasilkan benih. Perangkat lunak memproses hasilnya menjadi angka yang benar-benar acak dalam berbagai format. TRNG pertama yang tersedia secara komersial yang mencapai tingkat produksi bit yang sebanding dengan PRNG, adalah Intel digital random number generator (DRNG) [TAYL11], yang ditawarkan pada chip multicore baru sejak Mei 2012.