BAB 3

OTENTIKASI PENGGUNA

TUJUAN PEMBELAJARAN

Setelah mempelajari bab ini, Anda seharusnya dapat:

- Diskusikan empat cara umum untuk mengautentikasi identitas pengguna.
- Jelaskan mekanisme penggunaan kata sandi ter-hash untuk autentikasi pengguna.
- Memahami penggunaan filter Bloom dalam manajemen kata sandi.
- Menyajikan gambaran umum tentang autentikasi pengguna berbasis token.
- Diskusikan masalah yang terlibat dan pendekatan untuk autentikasi pengguna jarak jauh.
- Rangkuman beberapa masalah keamanan utama untuk autentikasi pengguna.

Dalam sebagian besar konteks keamanan komputer, autentikasi pengguna adalah blok bangunan fundamental dan garis pertahanan utama. Otentikasi pengguna adalah dasar untuk sebagian besar jenis kontrol akses dan akuntabilitas pengguna. RFC 4949 mendefinisikan autentikasi pengguna sebagai berikut:

Proses memverifikasi identitas yang diklaim oleh atau untuk entitas sistem. Proses autentikasi terdiri dari dua langkah:

- **Langkah identifikasi:** Memberikan pengenal ke sistem keamanan. (Pengidentifikasi harus diberikan dengan hati-hati, karena identitas yang diautentikasi adalah dasar untuk layanan keamanan lainnya, seperti layanan kontrol akses).
- **Langkah verifikasi:** Menyajikan atau menghasilkan informasi autentikasi yang menguatkan pengikatan antara entitas dan pengenal.

Sebagai contoh, pengguna Alice Toklas dapat memiliki pengenal pengguna ABTOKLAS. Informasi ini perlu disimpan di server atau sistem komputer mana pun yang ingin digunakan oleh Alice dan dapat diketahui oleh administrator sistem dan pengguna lain. Informasi autentikasi yang terkait dengan ID pengguna ini adalah kata sandi, yang dirahasiakan (hanya diketahui oleh Alice dan sistem)[1] . Jika tidak ada yang bisa mendapatkan atau menebak kata sandi Alice, maka kombinasi ID pengguna dan kata sandi Alice memungkinkan administrator untuk mengatur izin akses Alice dan mengaudit aktivitasnya. Karena ID Alice tidak rahasia, pengguna sistem dapat mengirimkan email kepadanya, tetapi karena kata sandinya rahasia, tidak ada yang dapat berpura-pura menjadi Alice.

Pada intinya, identifikasi adalah cara yang digunakan pengguna untuk memberikan identitas yang diklaim ke sistem; otentikasi pengguna adalah cara untuk menetapkan keabsahan klaim tersebut. Perhatikan bahwa autentikasi pengguna berbeda dengan autentikasi pesan. Seperti yang didefinisikan di Bab 2, otentikasi pesan adalah prosedur yang memungkinkan pihak-pihak yang berkomunikasi untuk memverifikasi bahwa isi pesan yang diterima belum diubah dan bahwa sumbernya asli. Bab ini hanya membahas autentikasi pengguna.

[1]Biasanya, kata sandi disimpan dalam bentuk hash di server dan kode hash ini mungkin tidak rahasia, seperti yang dijelaskan selanjutnya dalam bab ini.

Bab ini pertama-tama memberikan gambaran umum tentang berbagai cara autentikasi pengguna dan kemudian memeriksa masing-masing secara mendetail.

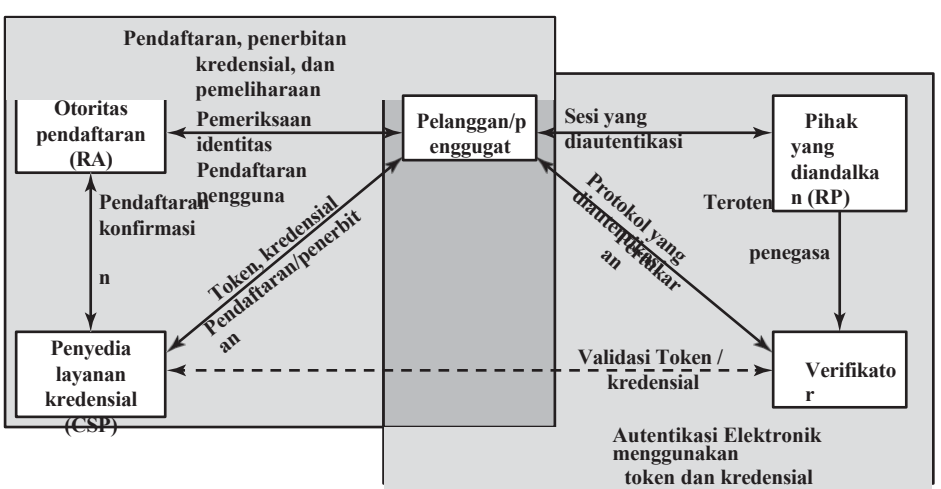
3. 1 Petunjuk Penggunaan Elektronik

NIST SP 800-63-2 (*Panduan Otentikasi Elektronik*, Agustus 2013) mendefinisikan otentikasi pengguna secara elektronik sebagai proses membangun kepercayaan pada identitas pengguna yang disajikan secara elektronik ke sistem informasi. Sistem dapat menggunakan identitas yang diautentikasi untuk menentukan apakah individu yang diautentikasi memiliki wewenang untuk melakukan fungsi tertentu, seperti transaksi basis data atau akses ke sumber daya sistem. Dalam banyak kasus, autentikasi dan transaksi atau fungsi resmi lainnya terjadi di jaringan terbuka seperti Internet. Otentikasi dan otorisasi selanjutnya dapat dilakukan secara lokal, seperti di jaringan area lokal.

Model untuk Otentikasi Pengguna Elektronik

SP 800-63-2 mendefinisikan model umum untuk autentikasi pengguna yang melibatkan sejumlah entitas dan prosedur. Kami membahas model ini dengan mengacu pada Gambar 3.1.

Persyaratan awal untuk melakukan autentikasi pengguna adalah bahwa pengguna harus terdaftar pada sistem. Berikut ini adalah urutan umum untuk pendaftaran. Pemohon mendaftar ke **otoritas pendaftaran (RA)** untuk menjadi **pelanggan** dari **penyedia layanan kepercayaan (CSP).** Dalam model ini, RA adalah entitas tepercaya yang menetapkan dan menjamin identitas pemohon kepada CSP. CSP kemudian melakukan pertukaran dengan pelanggan. Bergantung pada detail sistem otentikasi keseluruhan, CSP mengeluarkan semacam kredensial elektronik kepada pelanggan. **Kredensial** adalah struktur data yang secara otoritatif mengikat identitas dan atribut tambahan ke token yang dimiliki oleh pelanggan, dan dapat diverifikasi ketika dipresentasikan



Gambar 3.1 **Model Arsitektur Otentikasi Elektronik NIST SP 800-63-2**

kepada verifikator dalam transaksi otentikasi. Token dapat berupa kunci enkripsi atau kata sandi terenkripsi yang mengidentifikasi pelanggan. Token dapat dikeluarkan oleh CSP, dibuat langsung oleh pelanggan, atau disediakan oleh pihak ketiga. Token dan kredensial dapat digunakan dalam peristiwa otentikasi berikutnya.

Setelah pengguna terdaftar sebagai pelanggan, proses otentikasi yang sebenarnya dapat terjadi antara pelanggan dan satu atau beberapa sistem yang melakukan otentikasi dan, selanjutnya, otorisasi. Pihak yang akan diautentikasi disebut **pemohon** dan pihak yang memverifikasi identitas tersebut disebut **verifikator.** Ketika seorang pemohon berhasil menunjukkan kepemilikan dan kontrol token kepada verifier melalui protokol otentikasi, verifier dapat memverifikasi bahwa pemohon adalah pelanggan yang disebutkan dalam kredensial yang sesuai. Verifikator memberikan pernyataan tentang identitas pelanggan kepada **pihak yang mengandalkan (RP).** Pernyataan tersebut mencakup informasi identitas tentang pelanggan, seperti nama pelanggan, pengenal yang diberikan pada saat pendaftaran, atau atribut pelanggan lainnya yang diverifikasi dalam proses pendaftaran. RP dapat menggunakan informasi terotentikasi yang diberikan oleh verifikator untuk membuat keputusan kontrol akses atau otorisasi.

Sistem autentikasi yang diimplementasikan akan berbeda dari atau lebih rumit dari model yang disederhanakan ini, tetapi model ini menggambarkan peran dan fungsi utama yang diperlukan untuk sistem autentikasi yang aman.

Sarana Autentikasi

Ada empat cara umum untuk mengautentikasi identitas pengguna, yang dapat digunakan sendiri atau dikombinasikan:

- **Sesuatu yang diketahui individu:** Contohnya adalah kata sandi, nomor identifikasi pribadi (PIN), atau jawaban atas serangkaian pertanyaan yang telah diatur sebelumnya.
- **Sesuatu yang dimiliki oleh individu:** Contohnya termasuk kartu kunci elektronik, kartu pintar, dan kunci fisik. Jenis pengautentikasi ini disebut sebagai *token.*
- **Sesuatu yang dimiliki oleh individu (biometrik statis):** Contohnya termasuk pengenalan dengan sidik jari, retina, dan wajah.
- **Sesuatu yang dilakukan individu (biometrik dinamis):** Contohnya termasuk pengenalan melalui pola suara, karakteristik tulisan tangan, dan ritme mengetik.

Semua metode ini, jika diterapkan dan digunakan dengan benar, dapat memberikan autentikasi pengguna yang aman. Namun, setiap metode memiliki masalah. Musuh mungkin dapat menebak atau mencuri kata sandi. Demikian pula, musuh mungkin dapat memalsukan atau mencuri token. Seorang pengguna mungkin lupa kata sandi atau kehilangan token. Lebih lanjut, ada biaya administrasi yang signifikan untuk mengelola informasi kata sandi dan token pada sistem dan mengamankan informasi tersebut pada sistem. Sehubungan dengan pengautentikasi biometrik, t e r d a p a t berbagai masalah, termasuk berurusan dengan false positive dan false negative, penerimaan pengguna, biaya, dan kenyamanan.

Penilaian Risiko untuk Otentikasi Pengguna

Penilaian risiko keamanan secara umum dibahas di Bab 14. Di sini, kami memperkenalkan contoh spesifik yang berhubungan dengan autentikasi pengguna. Ada tiga hal yang terpisah

konsep yang ingin kami kaitkan satu sama lain: tingkat jaminan, potensi dampak, dan area risiko.

TINGKAT Kepastian Tingkat kepastian menggambarkan tingkat kepastian organisasi bahwa pengguna telah memberikan kredensial yang mengacu pada identitasnya. Secara lebih spesifik, jaminan didefinisikan sebagai (1) tingkat keyakinan dalam proses pemeriksaan yang digunakan untuk menetapkan identitas individu yang menerima kredensial dan (2) tingkat keyakinan bahwa individu yang menggunakan kredensial adalah individu yang menerima kredensial. SP 800-63-2 mengakui empat tingkat jaminan:

- **Level 1:** Sedikit atau tidak ada kepercayaan terhadap keabsahan identitas yang dinyatakan. Contoh di mana tingkat ini sesuai adalah konsumen yang mendaftar untuk berpartisipasi dalam diskusi di papan diskusi situs web perusahaan. Teknik otentikasi yang umum digunakan pada tingkat ini adalah ID dan kata sandi yang disediakan pengguna pada saat transaksi.
- **Tingkat 2:** Beberapa keyakinan terhadap keabsahan identitas yang dinyatakan. Kepercayaan tingkat 2 sesuai untuk berbagai macam bisnis dengan publik di mana organisasi memerlukan pernyataan identitas awal (rinciannya diverifikasi secara independen sebelum melakukan tindakan apa pun). Pada tingkat ini, beberapa jenis protokol autentikasi yang aman perlu digunakan, bersama dengan salah satu cara autentikasi yang telah dirangkum sebelumnya dan dibahas di bagian selanjutnya.
- **Level 3:** Keyakinan tinggi terhadap keabsahan identitas yang diberikan. Tingkat ini sesuai untuk memungkinkan klien atau karyawan mengakses layanan terbatas yang bernilai tinggi namun bukan yang tertinggi. Contoh yang sesuai dengan level ini: Seorang pengacara paten secara elektronik mengirimkan informasi paten rahasia ke Kantor Paten dan Merek Dagang AS. Pengungkapan yang tidak tepat akan memberikan keunggulan kompetitif bagi pesaing. Teknik yang perlu digunakan pada tingkat ini membutuhkan lebih dari satu faktor otentikasi; yaitu, setidaknya dua teknik otentikasi independen harus digunakan.
- **Tingkat 4:** Keyakinan yang sangat tinggi terhadap keabsahan identitas yang diberikan. Tingkat ini sesuai untuk memungkinkan klien atau karyawan mengakses layanan terbatas yang bernilai sangat tinggi atau yang jika diakses secara tidak benar akan sangat berbahaya. Sebagai contoh, seorang pejabat penegak hukum mengakses database penegakan hukum yang berisi catatan kriminal. Akses yang tidak sah dapat menimbulkan masalah privasi dan/atau membahayakan penyelidikan. Biasanya, autentikasi level 4 memerlukan penggunaan beberapa faktor serta pendaftaran secara langsung.

DAMPAK POTENSIAL Sebuah konsep yang terkait erat dengan tingkat jaminan adalah dampak potensial. FIPS 199 (*Standar Kategorisasi Keamanan Informasi dan Sistem Informasi Federal*, 2004) mendefinisikan tiga tingkat dampak potensial pada organisasi atau individu jika terjadi pelanggaran keamanan (dalam konteks kami, kegagalan dalam otentikasi pengguna):

- **Rendah:** Kesalahan autentikasi dapat diperkirakan memiliki efek buruk yang terbatas pada operasi organisasi, aset organisasi, atau individu. Secara lebih spesifik, kita dapat mengatakan bahwa kesalahan tersebut dapat: (1)

menyebabkan degradasi dalam misi

kemampuan sampai pada tingkat dan durasi tertentu sehingga organisasi dapat menjalankan fungsi utamanya, tetapi efektivitas fungsi tersebut berkurang secara nyata;
(2) mengakibatkan kerusakan kecil pada aset organisasi; (3) mengakibatkan kerugian finansial kecil pada organisasi atau individu; atau (4) mengakibatkan kerugian kecil pada individu.

- **Sedang:** Kesalahan otentikasi dapat diperkirakan akan menimbulkan dampak buruk yang serius. Lebih khusus lagi, kesalahan tersebut dapat: (1) menyebabkan penurunan yang signifikan dalam kemampuan misi sampai pada tingkat dan durasi dimana organisasi dapat menjalankan fungsi utamanya, namun efektivitas fungsi tersebut berkurang secara signifikan; (2) mengakibatkan kerusakan yang signifikan pada aset organisasi;
(3) mengakibatkan kerugian finansial yang signifikan; atau (4) mengakibatkan kerugian yang signifikan terhadap individu yang tidak melibatkan hilangnya nyawa atau cedera serius yang mengancam jiwa.
- **Tinggi:** Kesalahan otentikasi dapat diperkirakan akan menimbulkan dampak buruk yang parah atau sangat parah. Kesalahan tersebut dapat: (1) menyebabkan degradasi yang parah atau hilangnya kemampuan misi hingga pada tingkat dan durasi tertentu sehingga organisasi tidak dapat menjalankan satu atau lebih fungsi utamanya; (2) mengakibatkan kerusakan besar pada aset organisasi; (3) mengakibatkan kerugian finansial yang besar bagi organisasi atau individu; atau (4) mengakibatkan kerusakan parah atau bencana pada individu yang melibatkan hilangnya nyawa atau cedera serius yang mengancam jiwa.

AREA RISIKO Pemetaan antara potensi dampak dan tingkat asurans yang sesuai yang memuaskan untuk menangani potensi dampak tergantung pada konteksnya. Tabel 3.1 menunjukkan pemetaan yang memungkinkan untuk berbagai risiko yang mungkin dihadapi oleh organisasi. Tabel ini menunjukkan teknik untuk melakukan penilaian risiko. Untuk sistem informasi atau aset layanan tertentu dari sebuah organisasi, organisasi perlu menentukan tingkat dampak jika terjadi kegagalan otentikasi, dengan menggunakan kategori dampak, atau area risiko, yang menjadi perhatian.

Sebagai contoh, pertimbangkan potensi kerugian finansial jika terjadi kesalahan autentikasi yang mengakibatkan akses tidak sah ke basis data. Tergantung pada sifat database, dampaknya bisa berbeda:

- **Rendah:** Paling buruk, kerugian finansial yang tidak signifikan atau tidak penting yang tidak dapat dipulihkan kepada pihak mana pun, atau paling buruk, tanggung jawab organisasi yang tidak signifikan atau tidak penting.

Tabel 3.1 Potensi Dampak Maksimum untuk Setiap Tingkat Jaminan

	Profil Dampak Tingkat Jaminan			
Kategori Dampak Potensial untuk Kesalahan Otentikasi	**1**	**2**	**3**	**4**
Ketidaknyamanan, tekanan, atau kerusakan pada kedudukan atau reputasi	Renda h	Mod	Mod	Tinggi
Kerugian finansial atau kewajiban organisasi	Renda h	Mod	Mod	Tinggi
Membahayakan program atau kepentingan organisasi	Tidak	Renda	Mod	Tinggi

	ada	h		
Pelepasan informasi sensitif yang tidak sah	Tidak ada	Rendah	Mod	Tinggi
Keamanan pribadi	Tidak ada	Tidak ada	Rendah	Mod/ Tinggi
Pelanggaran perdata atau pidana	Tidak ada	Rendah	Mod	Tinggi

- **Sedang:** Paling buruk, kerugian finansial serius yang tidak dapat dipulihkan kepada pihak mana pun, atau tanggung jawab organisasi yang serius.
- **Tinggi:** kerugian finansial yang parah atau bencana yang tidak dapat dipulihkan kepada pihak mana pun; atau tanggung jawab organisasi yang parah atau bencana.

Tabel tersebut menunjukkan bahwa jika potensi dampaknya rendah, tingkat jaminan 1 sudah memadai. Jika potensi dampaknya sedang, tingkat jaminan 2 atau 3 harus dicapai. Dan jika potensi dampaknya tinggi, tingkat jaminan 4 harus diterapkan. Analisis serupa dapat dilakukan untuk kategori lain yang ditunjukkan dalam tabel. Analis kemudian dapat memilih tingkat jaminan yang memenuhi atau melampaui persyaratan untuk jaminan di setiap kategori yang tercantum dalam tabel. Jadi, misalnya, untuk sistem tertentu, jika salah satu kategori dampak memiliki potensi dampak tinggi, atau jika kategori keselamatan pribadi memiliki potensi dampak sedang atau tinggi, maka jaminan level 4 harus diterapkan.

3.2 OTENTIKASI BERBASIS PASSwOrD

Garis pertahanan yang banyak digunakan untuk melawan penyusup adalah sistem kata sandi. Hampir semua sistem multiuser, server berbasis jaringan, situs e-commerce berbasis Web, dan layanan serupa lainnya mengharuskan pengguna untuk memberikan tidak hanya nama atau pengenal (ID), tetapi juga kata sandi. Sistem membandingkan kata sandi dengan kata sandi yang telah disimpan sebelumnya untuk ID pengguna tersebut, yang disimpan dalam file kata sandi sistem. Kata sandi berfungsi untuk mengautentikasi ID individu yang masuk ke sistem. Pada gilirannya, ID memberikan keamanan dengan cara berikut:

- ID menentukan apakah pengguna berwenang untuk mendapatkan akses ke sistem. Pada beberapa sistem, hanya mereka yang telah memiliki ID yang diajukan pada sistem yang diizinkan untuk mendapatkan akses.
- ID menentukan hak istimewa yang diberikan kepada pengguna. Beberapa pengguna mungkin memiliki status pengawas atau "superuser" yang memungkinkan mereka untuk membaca file dan melakukan fungsi yang secara khusus dilindungi oleh sistem operasi. Beberapa sistem memiliki akun tamu atau anonim, dan pengguna akun ini memiliki hak istimewa yang lebih terbatas daripada yang lain.
- ID digunakan dalam apa yang disebut sebagai kontrol akses diskresioner. Sebagai contoh, dengan mencantumkan ID pengguna lain, seorang pengguna dapat memberikan izin kepada mereka untuk membaca file yang dimiliki oleh pengguna tersebut.

Kerentanan Kata Sandi

Pada subbagian ini, kami menguraikan bentuk serangan utama terhadap autentikasi berbasis kata sandi dan secara singkat menguraikan strategi penanggulangan. Bagian 3.2 selanjutnya akan membahas lebih detail tentang penanggulangan utama.

Biasanya, sistem yang menggunakan autentikasi berbasis kata sandi menyimpan file kata sandi yang diindeks oleh ID pengguna. Salah satu teknik yang biasanya digunakan adalah menyimpan bukan kata sandi pengguna tetapi fungsi hash satu arah dari kata sandi, seperti yang dijelaskan selanjutnya.

Kami dapat mengidentifikasi strategi serangan dan tindakan pencegahan berikut ini:

- **Serangan kamus offline:** Biasanya, kontrol akses yang kuat digunakan untuk melindungi file kata sandi sistem. Namun, pengalaman menunjukkan bahwa peretas yang nekad sering kali dapat menerobos kontrol tersebut dan mendapatkan akses ke file tersebut. Penyerang mendapatkan file kata sandi sistem dan membandingkan hash kata sandi dengan hash kata sandi yang umum digunakan. Jika ditemukan kecocokan, penyerang bisa mendapatkan akses dengan kombinasi ID/kata sandi tersebut. Tindakan penanggulangan meliputi kontrol untuk mencegah akses yang tidak sah ke file kata sandi, tindakan pendeteksian penyusupan untuk mengidentifikasi penyusupan, dan penerbitan ulang kata sandi secara cepat jika file kata sandi disusupi.
- **Serangan akun tertentu:** Penyerang menargetkan akun tertentu dan mengirimkan tebakan kata sandi sampai kata sandi yang benar ditemukan. Tindakan pencegahan standar adalah mekanisme penguncian akun, yang mengunci akses ke akun setelah beberapa kali percobaan login yang gagal. Praktik yang umum dilakukan adalah tidak lebih dari lima kali percobaan akses.
- **Serangan kata sandi populer:** Variasi dari serangan sebelumnya adalah menggunakan kata sandi populer dan mencobanya pada berbagai macam ID pengguna. Kecenderungan pengguna adalah memilih kata sandi yang mudah diingat; sayangnya hal ini membuat kata sandi mudah ditebak. Tindakan pencegahan termasuk kebijakan untuk menghambat pemilihan kata sandi umum oleh pengguna dan memindai alamat IP permintaan otentikasi dan cookie klien untuk mengetahui pola pengiriman.
- **Menebak kata sandi terhadap pengguna tunggal:** Penyerang mencoba untuk mendapatkan pengetahuan tentang pemegang akun dan kebijakan kata sandi sistem dan menggunakan pengetahuan tersebut untuk menebak kata sandi. Tindakan penanggulangannya meliputi pelatihan dan penegakan kebijakan kata sandi yang membuat kata sandi sulit ditebak. Kebijakan tersebut mencakup kerahasiaan, panjang minimum kata sandi, rangkaian karakter, larangan menggunakan pengenal pengguna yang terkenal, dan lama waktu sebelum kata sandi harus diubah.
- **Pembajakan stasiun kerja:** Penyerang menunggu sampai stasiun kerja yang login tidak dijaga. Penanggulangan standarnya adalah secara otomatis mengeluarkan stasiun kerja setelah beberapa saat tidak aktif. Skema deteksi penyusupan dapat digunakan untuk mendeteksi perubahan perilaku pengguna.
- **Mengeksploitasi kesalahan pengguna:** Jika sistem memberikan kata sandi, maka pengguna lebih cenderung menuliskannya karena sulit diingat. Situasi ini menciptakan potensi bagi pihak lawan untuk membaca kata sandi yang tertulis. Seorang pengguna mungkin dengan sengaja membagikan kata sandi, untuk memungkinkan seorang kolega berbagi file, misalnya. Selain itu, penyerang sering kali berhasil mendapatkan kata sandi dengan menggunakan taktik rekayasa sosial yang mengelabui pengguna atau manajer akun untuk mengungkapkan kata sandi. Banyak sistem komputer yang dikirimkan dengan kata sandi yang sudah dikonfigurasikan untuk administrator sistem. Kecuali jika kata sandi yang sudah dikonfigurasi ini diubah, kata sandi ini mudah ditebak. Tindakan pencegahannya meliputi pelatihan pengguna, deteksi penyusupan, dan kata sandi yang lebih sederhana yang

digabungkan dengan mekanisme autentikasi lain.

- **Mengeksploitasi penggunaan kata sandi ganda:** Serangan juga bisa menjadi jauh lebih efektif atau merusak jika perangkat jaringan yang berbeda berbagi kata sandi yang sama atau serupa.

kata sandi untuk pengguna tertentu. Tindakan pencegahan termasuk kebijakan yang melarang kata sandi yang sama atau mirip pada perangkat jaringan tertentu.

- **Pemantauan elektronik:** Jika kata sandi dikomunikasikan melalui jaringan untuk masuk ke sistem jarak jauh, maka kata sandi tersebut rentan terhadap penyadapan. Enkripsi sederhana tidak akan mengatasi masalah ini, karena kata sandi yang dienkripsi pada dasarnya adalah kata sandi yang dapat diamati dan digunakan kembali oleh pihak lawan.

Walaupun ada banyak kerentanan keamanan pada kata sandi, kata sandi tetap menjadi teknik autentikasi pengguna yang paling umum digunakan, dan hal ini sepertinya tidak akan berubah di masa yang akan datang [HERL12]. Di antara alasan popularitas kata sandi yang tetap bertahan adalah sebagai berikut:

1. Teknik yang menggunakan perangkat keras sisi klien, seperti pemindai sidik jari dan pembaca kartu pintar, memerlukan implementasi perangkat lunak autentikasi pengguna yang sesuai untuk mengeksploitasi perangkat keras ini pada sistem klien dan server. Sampai ada penerimaan yang luas di satu sisi, ada keengganan untuk mengimplementasikan di sisi lain, sehingga kita berakhir dengan kebuntuan siapa yang duluan.
2. Token fisik, seperti smart card, mahal dan/atau tidak nyaman untuk dibawa-bawa, terutama jika dibutuhkan banyak token.
3. Skema yang mengandalkan sistem masuk tunggal ke beberapa layanan, menggunakan salah satu teknik tanpa kata sandi yang dijelaskan dalam bab ini, menciptakan satu titik risiko keamanan.
4. Pengelola kata sandi otomatis yang meringankan pengguna dari beban mengetahui dan memasukkan kata sandi memiliki dukungan yang buruk untuk roaming dan sinkronisasi di berbagai platform klien, dan kegunaannya belum diteliti secara memadai.

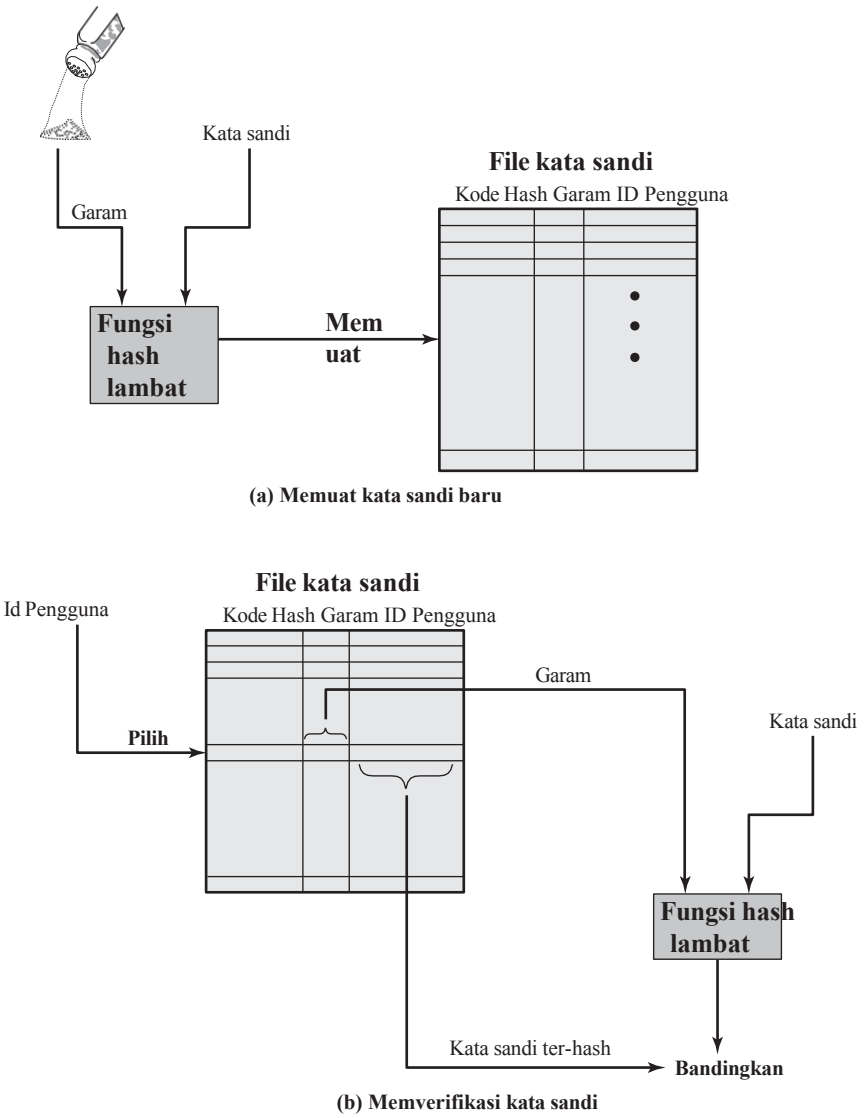
Oleh karena itu, ada baiknya kita mempelajari penggunaan kata sandi untuk autentikasi pengguna secara mendetail.

Penggunaan Kata Sandi Ter-hash

Teknik keamanan kata sandi yang banyak digunakan adalah penggunaan kata sandi ter-hash dan nilai garam. Skema ini ditemukan pada hampir semua varian UNIX dan juga sejumlah sistem operasi lainnya. Prosedur berikut ini digunakan (Gambar 3.2a). Untuk memasukkan password baru ke dalam sistem, pengguna memilih atau diberi password. Kata sandi ini digabungkan dengan nilai **garam** dengan panjang tetap [MORR79]. Pada implementasi yang lebih lama, nilai ini berhubungan dengan waktu di mana kata sandi diberikan kepada pengguna. Implementasi yang lebih baru menggunakan pseudorandom atau angka acak. Kata sandi dan salt berfungsi sebagai input untuk algoritma hashing untuk menghasilkan kode hash dengan panjang tetap. Algoritma hash dirancang untuk menjadi lambat untuk dieksekusi untuk menggagalkan serangan. Kata sandi hash kemudian disimpan, bersama dengan salinan plaintext dari salt, di dalam file kata sandi untuk ID pengguna yang bersangkutan. Metode kata sandi hash telah terbukti aman terhadap berbagai serangan kriptanalitik [WAGN00].

Ketika pengguna mencoba untuk masuk ke sistem UNIX, pengguna

memberikan ID dan password (Gambar 3.2b). Sistem operasi menggunakan ID tersebut untuk mengindeks ke dalam tabel



Gambar 3.2 Skema Kata Sandi UNIX

file kata sandi dan mengambil garam teks biasa dan kata sandi terenkripsi. Garam dan kata sandi yang disediakan pengguna digunakan sebagai masukan ke rutinitas enkripsi. Jika hasilnya cocok dengan nilai yang tersimpan, kata sandi diterima.

Garam memiliki tiga fungsi:

- Hal ini mencegah kata sandi duplikat terlihat di file kata sandi. Bahkan jika dua pengguna memilih kata sandi yang sama, kata sandi tersebut akan diberikan nilai salt yang berbeda. Oleh karena itu, kata sandi ter-hash dari kedua pengguna akan berbeda.

- Ini sangat meningkatkan kesulitan serangan kamus offline. Untuk garam dengan panjang *b* bit, jumlah kata sandi yang mungkin bertambah dengan faktor 2^b , meningkatkan kesulitan menebak kata sandi dalam serangan kamus.
- Hampir tidak mungkin untuk mengetahui apakah seseorang yang memiliki kata sandi di dua atau lebih sistem telah menggunakan kata sandi yang sama di semua sistem tersebut.

Untuk melihat poin kedua, pertimbangkan cara kerja serangan kamus offline. Penyerang mendapatkan salinan file kata sandi. Anggaplah terlebih dahulu bahwa salt tidak digunakan. Tujuan penyerang adalah untuk menebak satu kata sandi. Untuk itu, penyerang mengirimkan sejumlah besar kemungkinan kata sandi ke fungsi hashing. Jika salah satu tebakan cocok dengan salah satu hash di dalam berkas, maka penyerang telah menemukan kata sandi yang ada di dalam berkas tersebut. Tetapi dihadapkan pada skema UNIX, penyerang harus mengambil setiap tebakan dan mengirimkannya ke fungsi hash satu kali untuk setiap nilai salt di file kamus, mengalikan jumlah tebakan yang harus diperiksa.

Ada dua ancaman terhadap skema kata sandi UNIX. Pertama, seorang pengguna bisa mendapatkan akses ke sebuah mesin dengan menggunakan akun tamu atau dengan cara lain dan kemudian menjalankan sebuah program untuk menebak kata sandi, yang disebut password cracker, pada mesin tersebut. Penyerang seharusnya dapat memeriksa ribuan kemungkinan kata sandi dengan konsumsi sumber daya yang sedikit. Sebagai tambahan, jika lawan bisa mendapatkan salinan file kata sandi, maka program cracker bisa dijalankan pada mesin lain di waktu luang. Hal ini memungkinkan lawan untuk memeriksa jutaan kemungkinan kata sandi dalam waktu yang wajar.

IMPLEMENTASI UNIX Sejak awal pengembangan UNIX, sebagian besar implementasi mengandalkan skema kata sandi berikut ini. Setiap pengguna memilih kata sandi hingga delapan karakter yang dapat dicetak. Ini dikonversi menjadi nilai 56-bit (menggunakan ASCII 7-bit) yang berfungsi sebagai input kunci untuk rutinitas enkripsi. Rutinitas hash, yang dikenal sebagai crypt(3), didasarkan pada DES. Sebuah nilai garam 12-bit digunakan. Algoritma DES yang dimodifikasi dieksekusi dengan input data yang terdiri dari sebuah blok 64-bit angka nol. Output dari algoritma ini kemudian digunakan sebagai input untuk enkripsi kedua. Proses ini diulangi untuk total 25 enkripsi. Output 64-bit yang dihasilkan kemudian diterjemahkan ke dalam urutan 11 karakter. Modifikasi algoritma DES mengubahnya menjadi sebuah fungsi hash satu arah. Rutin crypt(3) didesain untuk mencegah serangan tebak-tebakan. Implementasi perangkat lunak dari DES lebih lambat dibandingkan dengan versi perangkat kerasnya, dan penggunaan 25 kali iterasi melipatgandakan waktu yang dibutuhkan sebanyak 25 kali.

Implementasi khusus ini sekarang dianggap sangat tidak memadai. Sebagai contoh, [PERR03] melaporkan hasil dari serangan kamus menggunakan superkomputer. Serangan ini mampu memproses lebih dari 50 juta tebakan kata sandi dalam waktu sekitar 80 menit. Lebih lanjut, hasil penelitian menunjukkan bahwa dengan biaya sekitar $10.000, siapa pun dapat melakukan hal yang sama dalam beberapa bulan dengan menggunakan satu mesin uniprosesor. Meskipun sudah diketahui kelemahannya, skema UNIX ini masih sering dibutuhkan untuk kompatibilitas dengan perangkat lunak manajemen akun yang ada atau dalam lingkungan multivendor.

Ada skema hash/salt lain yang lebih kuat yang tersedia untuk UNIX. Fungsi

hash yang direkomendasikan untuk banyak sistem UNIX, termasuk Linux, Solaris, dan FreeBSD (UNIX sumber terbuka yang banyak digunakan), didasarkan pada algoritme hash aman MD5 (yang mirip dengan, tetapi tidak seaman SHA-1). The

Rutinitas kriptografi MD5 menggunakan garam hingga 48 bit dan secara efektif tidak memiliki batasan panjang kata sandi. Ini menghasilkan nilai hash 128-bit. Ini juga jauh lebih lambat daripada crypt(3). Untuk mencapai perlambatan tersebut, MD5 crypt menggunakan sebuah putaran dalam dengan 1000 iterasi.

Mungkin versi yang paling aman dari skema hash/salt UNIX dikembangkan untuk OpenBSD, sebuah UNIX open source yang banyak digunakan. Skema ini, yang dilaporkan pada [PROV99], menggunakan fungsi hash yang berdasarkan pada sandi blok simetris Blowfish. Fungsi hash, yang disebut Bcrypt, cukup lambat untuk dieksekusi. Bcrypt mengizinkan kata sandi dengan panjang hingga 55 karakter dan membutuhkan nilai salt acak sebesar 128 bit, untuk menghasilkan nilai hash 192-bit. Bcrypt juga menyertakan sebuah variabel biaya; peningkatan pada variabel biaya akan menyebabkan peningkatan waktu yang dibutuhkan untuk melakukan hash Bcyrpt. Biaya yang diberikan pada kata sandi baru dapat dikonfigurasi, sehingga administrator dapat memberikan biaya yang lebih tinggi pada pengguna yang memiliki hak istimewa.

Pembobolan Kata Sandi dari Kata Sandi yang Dipilih Pengguna

PENDEKATAN TRADISIONAL Pendekatan tradisional untuk menebak kata sandi, atau biasa disebut dengan password cracking, adalah dengan mengembangkan sebuah kamus besar kata sandi yang mungkin dan mencoba setiap kata sandi tersebut pada file kata sandi. Ini berarti setiap kata sandi harus di-hash menggunakan setiap nilai garam yang tersedia dan kemudian dibandingkan dengan nilai hash yang tersimpan. Jika tidak ada kecocokan yang ditemukan, program cracking akan mencoba variasi pada semua kata dalam kamus kata sandi yang mungkin. Variasi tersebut termasuk pengejaan kata yang terbalik, angka tambahan atau karakter khusus, atau urutan karakter.

Sebuah alternatif adalah menukar ruang dengan waktu dengan melakukan prakomputasi nilai hash potensial. Dalam pendekatan ini, penyerang membuat sebuah kamus besar yang berisi kata-kata sandi yang mungkin. Untuk setiap kata sandi, penyerang menghasilkan nilai hash yang terkait dengan setiap kemungkinan nilai salt. Hasilnya adalah sebuah tabel besar nilai hash yang dikenal sebagai **tabel** pelangi. Sebagai contoh, [OECH03] menunjukkan bahwa dengan menggunakan 1.4 GB data, dia dapat memecahkan 99.9% dari semua hash kata sandi Windows alfanumerik dalam 13.8 detik. Pendekatan ini dapat dilawan dengan menggunakan nilai salt yang cukup besar dan panjang hash yang cukup besar. Pendekatan FreeBSD dan OpenBSD seharusnya aman dari serangan ini di masa mendatang.

Untuk melawan penggunaan nilai salt dan panjang hash yang besar, para peretas kata sandi mengeksploitasi fakta bahwa beberapa orang memilih kata sandi yang mudah ditebak. Beberapa pengguna, ketika diizinkan untuk memilih kata sandi mereka sendiri, memilih kata sandi yang sangat pendek. Sebuah penelitian di Universitas Purdue [SPAF92a] mengamati pilihan penggantian kata sandi pada 54 mesin, yang mewakili sekitar 7000 akun pengguna. Hampir 3% dari kata sandi tersebut terdiri dari tiga karakter atau kurang dari itu. Seorang penyerang dapat memulai serangan dengan menguji semua kemungkinan kata sandi yang panjangnya 3 karakter atau kurang. Solusi sederhana adalah dengan membuat sistem menolak semua pilihan kata sandi yang kurang dari, katakanlah, enam karakter atau bahkan mengharuskan semua kata sandi memiliki panjang delapan karakter. Sebagian besar pengguna tidak akan mengeluhkan pembatasan seperti

itu.

Panjang kata sandi hanyalah sebagian dari masalah. Banyak orang, ketika d i i z i n k a n untuk memilih kata sandi mereka sendiri, memilih kata sandi yang mudah ditebak, seperti nama mereka sendiri, nama jalan mereka, kata dalam kamus, dan lain sebagainya. Hal ini membuat pekerjaan pembobolan kata sandi menjadi mudah. Pembobol hanya perlu menguji

kata sandi terhadap daftar kata sandi yang mungkin. Karena banyak orang menggunakan kata sandi yang mudah ditebak, strategi seperti ini seharusnya berhasil pada hampir semua sistem.

Salah satu demonstrasi dari keefektifan menebak dilaporkan dalam [KLEI90]. Dari berbagai sumber, penulis mengumpulkan file kata sandi UNIX, yang berisi hampir 14.000 kata sandi terenkripsi. Hasilnya, yang oleh penulisnya sendiri dikategorikan sebagai menakutkan, adalah bahwa secara keseluruhan, hampir seperempat dari password tersebut dapat ditebak. Strategi berikut ini digunakan:

1. Cobalah nama pengguna, inisial, nama akun, dan informasi pribadi lainnya yang relevan. Secara keseluruhan, 130 permutasi yang berbeda untuk setiap pengguna telah dicoba.
2. Cobalah kata-kata dari berbagai kamus. Penulis menyusun kamus yang berisi lebih dari 60.000 kata, termasuk kamus online pada sistem itu sendiri, dan berbagai daftar lain seperti yang ditunjukkan.
3. Cobalah berbagai permutasi pada kata-kata dari langkah 2. Ini termasuk membuat huruf pertama menjadi huruf besar atau karakter kontrol, membuat seluruh kata menjadi huruf besar, membalikkan kata, mengubah huruf "o" menjadi angka "nol", dan seterusnya. Permutasi ini menambahkan 1 juta kata lagi ke dalam daftar.
4. Cobalah berbagai permutasi huruf besar pada kata-kata dari langkah 2 yang tidak dipertimbangkan pada langkah 3. Hal ini menambahkan hampir 2 juta kata tambahan ke dalam daftar.

Dengan demikian, pengujian ini melibatkan hampir 3 juta kata. Dengan menggunakan prosesor tercepat yang tersedia, waktu yang dibutuhkan untuk mengenkripsi semua kata tersebut untuk semua nilai salt yang mungkin adalah kurang dari satu jam. Perlu diingat bahwa pencarian menyeluruh seperti itu dapat menghasilkan tingkat keberhasilan sekitar 25%, sedangkan satu serangan saja sudah cukup untuk mendapatkan berbagai macam hak istimewa pada sebuah sistem.

Serangan yang menggunakan kombinasi teknik brute-force dan kamus sudah menjadi hal yang umum. Contoh penting dari pendekatan ganda ini adalah John the Ripper, peretas kata sandi sumber terbuka yang pertama kali dikembangkan pada tahun 1996 dan masih digunakan [OPEN13].

Pendekatan Modern Sayangnya, kerentanan jenis ini tidak berkurang dalam 25 tahun terakhir ini. Para pengguna melakukan pekerjaan yang lebih baik dalam memilih kata sandi, dan organisasi melakukan pekerjaan yang lebih baik dalam memaksa para pengguna untuk memilih kata sandi yang lebih kuat, sebuah konsep yang dikenal sebagai kebijakan kata sandi yang kompleks, seperti yang akan dibahas selanjutnya. Namun, teknik pembobolan kata sandi telah meningkat untuk mengimbangi. Peningkatannya terdiri dari dua jenis. Pertama, kapasitas pemrosesan yang tersedia untuk memecahkan kata sandi telah meningkat secara dramatis. Sekarang semakin banyak digunakan untuk komputasi, prosesor grafis memungkinkan program pembobol kata sandi bekerja ribuan kali lebih cepat daripada yang mereka lakukan hanya satu dekade yang lalu pada PC dengan harga yang sama yang menggunakan CPU tradisional saja. Sebuah PC yang menjalankan satu GPU AMD Radeon HD7970, misalnya, dapat mencoba rata-rata sebuah

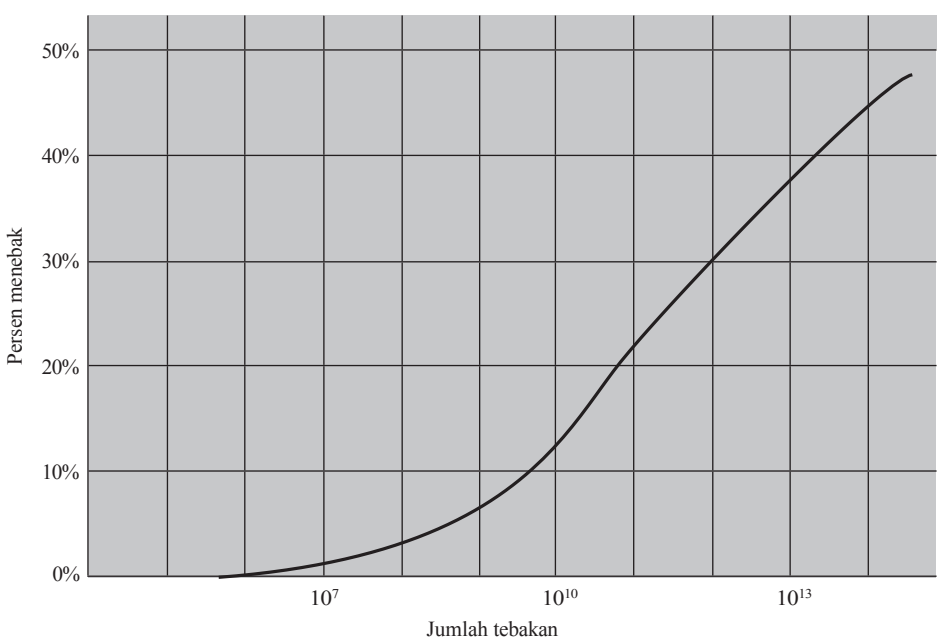
8.2 10^9 kombinasi kata sandi setiap detiknya, tergantung pada algoritma yang digunakan untuk mengacak kata sandi [GOOD12a]. Hanya satu dekade yang lalu, kecepatan seperti itu hanya mungkin dicapai jika menggunakan superkomputer yang mahal.

Area kedua dari peningkatan dalam pembobolan kata sandi adalah penggunaan algoritma canggih untuk menghasilkan kata sandi yang potensial. Sebagai contoh, [NARA05] mengembangkan sebuah model untuk pembuatan kata sandi dengan menggunakan probabilitas huruf dalam bahasa alami. Para peneliti menggunakan teknik pemodelan Markov standar dari pemrosesan bahasa alami untuk secara dramatis mengurangi ukuran ruang kata sandi yang akan dicari.

Tetapi hasil terbaik telah dicapai dengan mempelajari contoh-contoh kata sandi yang sebenarnya digunakan. Untuk mengembangkan teknik yang lebih efisien dan efektif daripada serangan kamus sederhana dan brute-force, para peneliti dan peretas telah mempelajari struktur kata sandi. Untuk melakukan hal ini, para analis membutuhkan kumpulan kata sandi yang besar untuk dipelajari, yang sekarang sudah mereka miliki. Terobosan besar pertama terjadi pada akhir 2009, ketika sebuah serangan injeksi SQL terhadap layanan game online RockYou.com mengekspos 32 juta kata sandi plaintext yang digunakan oleh para anggotanya untuk masuk ke akun mereka [TIMM10]. Sejak saat itu, banyak kumpulan file kata sandi yang bocor telah tersedia untuk dianalisis.

Dengan menggunakan set data besar kata sandi yang bocor sebagai data pelatihan, [WEIR09] melaporkan pengembangan tata bahasa bebas konteks probabilistik untuk pembobolan kata sandi. Dalam pendekatan ini, tebakan-tebakan diurutkan menurut kemungkinannya, berdasarkan pada frekuensi struktur kelas karakter dalam data pelatihan, serta frekuensi substring digit dan simbolnya. Pendekatan ini telah terbukti efisien dalam memecahkan kata sandi [KELL12, ZHAN10].

[MAZU13] melaporkan sebuah analisis dari kata sandi yang digunakan oleh lebih dari 25.000 mahasiswa di sebuah universitas riset dengan kebijakan kata sandi yang rumit. Para analis menggunakan pendekatan pemecahan kata sandi yang diperkenalkan di [WEIR09]. Mereka menggunakan basis data yang terdiri dari koleksi file kata sandi yang bocor, termasuk file RockYou. Gambar 3.3 merangkum hasil utama dari makalah tersebut. Grafik tersebut menunjukkan persentase kata sandi yang telah dipulihkan sebagai fungsi dari jumlah tebakan. Seperti yang dapat dilihat, lebih dari 10% kata sandi dipulihkan setelah hanya 10 tebakan^{10} . Setelah 10 tebakan^{13} , hampir 40% kata sandi dipulihkan.



Gambar 3.3 Persentase Kata Sandi yang Ditebak Setelah Sejumlah Tebakan Tertentu

Kontrol Akses File Kata Sandi

Salah satu cara untuk menggagalkan serangan kata sandi adalah dengan menolak akses lawan ke file kata sandi. Jika bagian kata sandi ter-hash dari file tersebut hanya dapat diakses oleh pengguna yang memiliki hak istimewa, maka lawan tidak dapat membacanya tanpa mengetahui kata sandi pengguna yang memiliki hak istimewa. Sering kali, kata sandi ter-hash disimpan di file terpisah dari ID pengguna, yang disebut sebagai **file** kata **sandi bayangan.** Perhatian khusus diberikan untuk membuat file kata sandi bayangan terlindungi dari akses yang tidak sah. Walaupun proteksi file kata sandi tentu saja bermanfaat, namun tetap ada kerentanan:

- Banyak sistem, termasuk sebagian besar sistem UNIX, rentan terhadap pembobolan yang tidak diantisipasi. Seorang peretas mungkin dapat mengeksploitasi kerentanan perangkat lunak dalam sistem operasi untuk mem-bypass sistem kontrol akses yang cukup lama untuk mengekstrak file kata sandi. Atau, peretas dapat menemukan kelemahan pada sistem berkas atau sistem manajemen basis data yang memungkinkan akses ke berkas tersebut.
- Kecelakaan proteksi dapat membuat file kata sandi dapat dibaca, sehingga menjanjikan semua akun.
- Beberapa pengguna memiliki akun di mesin lain di domain proteksi lain, dan mereka menggunakan kata sandi yang sama. Jadi, jika kata sandi dapat dibaca oleh siapa pun di satu mesin, mesin di lokasi lain bisa saja disusupi.
- Kurangnya atau lemahnya keamanan fisik dapat memberikan peluang bagi peretas. Kadang-kadang ada cadangan file kata sandi pada disk perbaikan darurat atau disk arsip. Akses ke cadangan ini memungkinkan penyerang untuk membaca file kata sandi. Sebagai alternatif, pengguna dapat melakukan booting dari disk yang menjalankan sistem operasi lain seperti Linux dan mengakses file dari OS ini.
- Alih-alih mengambil file kata sandi sistem, pendekatan lain untuk mengumpulkan ID dan kata sandi pengguna adalah dengan mengendus lalu lintas jaringan.

Dengan demikian, kebijakan perlindungan kata sandi harus melengkapi langkah-langkah kontrol akses dengan teknik untuk memaksa pengguna memilih kata sandi yang sulit ditebak.

Strategi Pemilihan Kata Sandi

Jika tidak dibatasi, banyak pengguna memilih kata sandi yang terlalu pendek atau terlalu mudah ditebak. Di sisi lain, jika pengguna diberikan kata sandi yang terdiri dari delapan karakter yang dapat dicetak yang dipilih secara acak, pembobolan kata sandi secara efektif tidak mungkin dilakukan. Tetapi hampir mustahil bagi sebagian besar pengguna untuk mengingat kata sandi mereka. Untungnya, bahkan jika kita membatasi semesta kata sandi pada deretan karakter yang cukup mudah diingat, ukuran semesta ini masih terlalu besar untuk memungkinkan pembobolan praktis. Tujuan kami adalah untuk menghilangkan kata sandi yang mudah ditebak dan mengizinkan pengguna untuk memilih kata sandi yang mudah diingat. Ada empat teknik dasar yang digunakan:

- Pendidikan pengguna

- Kata sandi yang dibuat oleh komputer
- Pemeriksaan kata sandi reaktif
- Kebijakan kata sandi yang rumit

Pengguna bisa diberitahu pentingnya menggunakan kata sandi yang sulit ditebak dan bisa diberikan panduan untuk memilih kata sandi yang kuat. Strategi **pendidikan pengguna** ini mungkin tidak akan berhasil pada sebagian besar instalasi, terutama di mana terdapat populasi pengguna yang besar atau banyak pergantian. Banyak pengguna yang akan mengabaikan panduan tersebut. Sebagian lainnya mungkin tidak dapat menilai dengan baik apa yang dimaksud dengan kata sandi yang kuat. Sebagai contoh, banyak pengguna (secara keliru) percaya bahwa membalikkan kata atau menggunakan huruf besar pada huruf terakhir akan membuat kata sandi tidak bisa ditebak.

Meskipun demikian, masuk akal untuk menyediakan panduan bagi pengguna tentang pemilihan kata sandi. Mungkin pendekatan terbaik adalah saran berikut ini: Teknik yang baik untuk memilih kata sandi adalah dengan menggunakan huruf pertama dari setiap kata dalam sebuah frasa. Namun, jangan memilih frasa yang sudah terkenal seperti "Sebuah apel sehari membuat dokter pergi" (Aaadktda). Sebaliknya, pilihlah sesuatu seperti "Nama depan anjing saya adalah Rex" (MdfniR) atau "Kakak perempuan saya, Peg, berumur 24 tahun" (MsPi24yo). Penelitian telah menunjukkan bahwa pengguna pada umumnya dapat mengingat kata sandi seperti itu, tetapi mereka tidak rentan terhadap serangan menebak kata sandi berdasarkan kata sandi yang umum digunakan.

Kata sandi yang dibuat oleh komputer juga memiliki masalah. Jika kata sandi cukup acak, pengguna tidak akan dapat mengingatnya. Bahkan jika kata sandi dapat diucapkan, pengguna mungkin akan kesulitan mengingatnya sehingga tergoda untuk menuliskannya. Secara umum, skema kata sandi yang dibuat oleh komputer memiliki sejarah penerimaan yang buruk oleh pengguna. FIPS 181 mendefinisikan salah satu pembuat kata sandi otomatis yang dirancang paling baik. Standar ini tidak hanya mencakup deskripsi pendekatan tetapi juga daftar lengkap kode sumber C dari algoritmanya. Algoritme ini menghasilkan kata-kata dengan membentuk suku kata yang dapat diucapkan dan menggabungkannya untuk membentuk sebuah kata. Generator angka acak menghasilkan aliran karakter acak yang digunakan untuk membangun suku kata dan kata.

Strategi **pengecekan kata sandi reaktif** adalah strategi di mana sistem secara berkala menjalankan peretas kata sandi sendiri untuk menemukan kata sandi yang dapat ditebak. Sistem ini dapat menyimpan kata sandi apa pun yang dapat ditebak dan memberi tahu pengguna. Taktik ini memiliki beberapa kelemahan. Pertama, ini membutuhkan banyak sumber daya jika pekerjaan dilakukan dengan benar. Karena lawan yang gigih dan mampu mencuri file kata sandi dapat mencurahkan waktu CPU penuh untuk tugas itu selama berjam-jam atau bahkan berhari-hari, pemeriksa kata sandi reaktif yang efektif memiliki kelemahan yang jelas. Lebih jauh lagi, semua kata sandi yang ada tetap rentan sampai pemeriksa kata sandi reaktif menemukannya. Sebuah contoh yang bagus adalah peretas kata sandi openware Jack the Ripper (openwall.com/john/pro/), yang bekerja pada berbagai sistem operasi.

Pendekatan yang menjanjikan untuk meningkatkan keamanan kata sandi adalah **kebijakan kata sandi** yang **kompleks**, atau pemeriksa **kata sandi proaktif.** Dalam skema ini, pengguna diizinkan untuk memilih kata sandinya sendiri. Namun, pada saat pemilihan, sistem akan memeriksa apakah kata sandi tersebut diperbolehkan dan, jika tidak, akan menolaknya. Pemeriksa seperti ini berdasarkan pada phi losophy bahwa, dengan panduan yang cukup dari sistem, pengguna bisa memilih kata sandi yang mudah diingat dari ruang kata sandi yang cukup besar yang tidak mungkin bisa ditebak dalam serangan kamus.

Trik dengan pemeriksa kata sandi proaktif adalah mencapai keseimbangan

antara penerimaan dan kekuatan pengguna. Jika sistem menolak terlalu banyak kata sandi, pengguna akan mengeluh karena terlalu sulit untuk memilih kata sandi. Jika sistem menggunakan beberapa algoritme sederhana untuk menentukan apa yang bisa diterima, ini memberikan panduan bagi para pembobol kata sandi

untuk menyempurnakan teknik tebakan mereka. Pada bagian selanjutnya dari subbab ini, kita akan membahas pendekatan yang memungkinkan untuk memeriksa kata sandi secara proaktif.

PENEGAKAN ATURAN Pendekatan pertama adalah sistem sederhana untuk penegakan aturan. Sebagai contoh, aturan berikut ini dapat ditegakkan:

- Semua kata sandi harus terdiri dari minimal delapan karakter.
- Dalam delapan karakter pertama, kata sandi harus menyertakan setidaknya satu huruf besar, huruf kecil, angka, dan tanda baca.

Peraturan-peraturan ini dapat digabungkan dengan saran pada pengguna. Walaupun pendekatan ini lebih unggul daripada sekadar mendidik pengguna, namun mungkin tidak cukup untuk menggagalkan pembobol kata sandi. Skema ini memperingatkan para peretas tentang kata sandi mana yang *tidak* boleh dicoba tetapi masih memungkinkan untuk melakukan pembobolan kata sandi.

Proses penegakan aturan dapat diotomatisasi dengan menggunakan pemeriksa kata sandi proaktif, seperti openware pam_passwdqc (openwall.com/passwdqc/), yang menegakkan berbagai aturan tentang kata sandi dan dapat dikonfigurasi oleh administrator sistem.

PEMERIKSA KATA SANDI Prosedur lain yang mungkin dilakukan adalah dengan menyusun kamus besar kata sandi yang mungkin "buruk". Ketika pengguna memilih kata sandi, sistem akan memeriksa untuk memastikan bahwa kata sandi tersebut tidak ada dalam daftar yang tidak disetujui. Ada dua masalah dengan pendekatan ini:

- **Ruang:** Kamus harus berukuran sangat besar agar efektif. Sebagai contoh, kamus yang digunakan dalam studi Purdue [SPAF92a] menempati lebih dari 30 MB penyimpanan.
- **Waktu: Waktu yang** diperlukan untuk mencari kamus yang besar, bisa jadi sangat lama. Selain itu, untuk memeriksa kemungkinan permutasi kata-kata kamus, kata-kata tersebut harus disertakan dalam kamus, membuatnya benar-benar besar, atau setiap pencarian juga harus melibatkan pemrosesan yang cukup besar.

FILTER BLOOM Sebuah teknik [SPAF92a, SPAF92b] untuk mengembangkan pemeriksa kata sandi proaktif yang efektif dan efisien yang didasarkan pada penolakan kata-kata pada daftar telah diimplementasikan pada sejumlah sistem, termasuk Linux. Hal ini didasarkan pada penggunaan filter Bloom [BLOO70]. Untuk memulainya, kami akan menjelaskan pengoperasian filter Bloom. Filter Bloom dengan orde k terdiri dari sekumpulan k fungsi hash independen $H_1(x)$, $H_2(x)$, c, $H_k(x)$, di mana setiap fungsi memetakan kata sandi ke dalam sebuah nilai hash di dalam rentang 0 sampai N - 1. Artinya,

$$H_i(X_j) = y \qquad 1 \ldots i \ldots k; \qquad 1 \ldots j \ldots d \quad 0 \ldots y \ldots N - 1$$

di mana

X_j kata ke-j dalam kamus kata sandi

D jumlah kata dalam kamus kata sandi

Prosedur berikut ini kemudian diterapkan pada kamus:

1. Tabel hash dengan N bit didefinisikan, dengan semua bit pada awalnya disetel ke 0.
2. Untuk setiap kata sandi, nilai hash *k-nya* dihitung, dan bit yang sesuai pada tabel hash diatur ke 1. Jadi, jika H_i (X_j) 67 untuk beberapa (i, j), maka bit keenam puluh tujuh pada tabel hash diatur ke 1; jika bit tersebut telah memiliki nilai 1, maka bit tersebut akan tetap berada di angka 1.

Ketika kata sandi baru diberikan kepada pemeriksa, nilai k hash-nya dihitung. Jika semua bit yang sesuai pada tabel hash sama dengan 1, maka kata sandi ditolak. Semua kata sandi dalam kamus akan ditolak. Tetapi akan ada juga beberapa "false positive" (yaitu kata sandi yang tidak ada di dalam kamus tetapi menghasilkan kecocokan pada tabel hash). Untuk melihat hal ini, pertimbangkan skema dengan dua fungsi hash. Misalkan kata sandi *undertaker* dan *hulkhogan* ada di dalam kamus, tetapi *xG%# jj98* tidak. Lebih jauh lagi, anggaplah bahwa

H_1 (pengelola)	25	H_1 (hulkhogan)	83	H_1 (xG%#jj98)	665
H_2 (pengelola)	998	H_2 (hulkhogan)	665	H_2 (xG%#jj98)	998

Jika kata sandi xG%#jj98 diberikan kepada sistem, maka akan ditolak meskipun tidak ada dalam kamus. Jika ada terlalu banyak kesalahan positif, akan sulit bagi pengguna untuk memilih kata sandi. Oleh karena itu, kami ingin mendesain skema hash untuk meminimalisir false positive. Dapat ditunjukkan bahwa probabilitas P dari sebuah false positive dapat didekati dengan

$$P \approx 11 - e\text{-}kD/N^{2k} = 11 - e\text{-}k/R^{2k}$$

atau, secara ekuivalen,

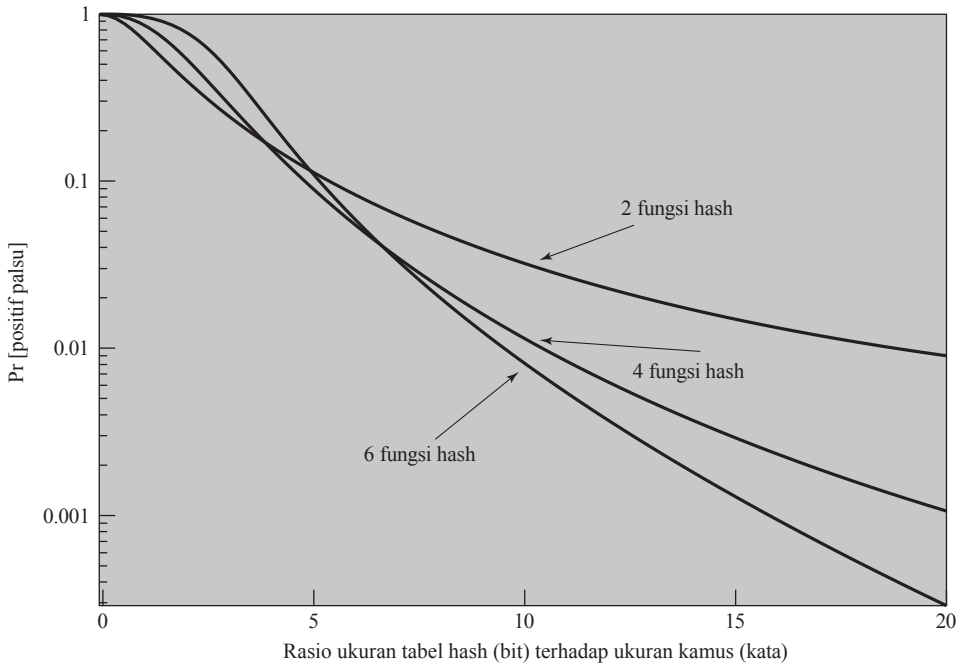
$$R \approx \frac{-k}{\ln(1 - p)^{1/k}}$$

di mana
- k jumlah fungsi hash
- N jumlah bit dalam tabel hash
- D jumlah kata dalam kamus
- R N/D, rasio ukuran tabel hash (bit) terhadap ukuran kamus (kata)

Gambar 3.4 memplot P sebagai fungsi dari R untuk berbagai nilai k. Misalkan kita memiliki sebuah kamus dengan 1 juta kata dan kita ingin memiliki probabilitas 0.01 untuk menolak kata sandi yang tidak ada di dalam kamus. Jika kita memilih enam fungsi hash, rasio yang dibutuhkan adalah R 9.6. Oleh karena itu, kita membutuhkan tabel hash sebesar 9.6 10^6 bit atau sekitar 1.2 MB penyimpanan. Sebaliknya, penyimpanan seluruh kamus akan membutuhkan sekitar 8 MB. Dengan demikian, kita mencapai kompresi hampir satu faktor dari 7. Lebih jauh lagi, pengecekan kata sandi melibatkan perhitungan langsung dari enam fungsi hash dan tidak bergantung pada ukuran kamus, sedangkan dengan penggunaan kamus lengkap, ada pencarian yang cukup besar.[2]

[2]Filter Bloom melibatkan penggunaan teknik probabilistik. Ada kemungkinan kecil bahwa beberapa kata sandi yang tidak ada di dalam kamus akan ditolak. Sering kali dalam mendesain algoritma, penggunaan

teknik probabilistik menghasilkan solusi yang tidak terlalu memakan waktu atau tidak terlalu rumit, atau keduanya.



Gambar 3.4 Performa Filter Bloom

3.3 AUTENTIKASI BERBASIS TOKEN

Benda yang dimiliki oleh pengguna untuk tujuan otentikasi pengguna disebut token. Pada bagian ini, kita akan membahas dua jenis token yang banyak digunakan, yaitu kartu yang memiliki tampilan dan ukuran seperti kartu bank (lihat Tabel 3.2).

Kartu Memori

Kartu memori dapat menyimpan tetapi tidak dapat memproses data. Kartu yang paling umum adalah kartu bank dengan strip magnetik di bagian belakangnya. Strip magnetik hanya dapat menyimpan kode keamanan sederhana, yang dapat dibaca (dan sayangnya diprogram ulang) oleh pembaca kartu yang tidak mahal. Ada juga kartu memori yang menyertakan memori elektronik internal.

Tabel 3.2 Jenis Kartu yang Digunakan sebagai Token

Jenis Kartu	Fitur Penentu	Contoh
Timbul	Hanya karakter yang dibesarkan, di bagian depan	Kartu kredit lama
Strip magnetik	Bilah magnet di belakang, karakter di depan	Kartu bank
Memori	Memori elektronik di dalam	Kartu telepon prabayar

Kontak	Memori elektronik dan prosesor di dalam	Kartu identitas biometrik
Pintar Tanpa Kontak	Kontak listrik yang terpapar di permukaan Antena radio tertanam di dalam	

Kartu memori dapat digunakan sendiri untuk akses fisik, seperti kamar hotel. Untuk autentikasi, pengguna memberikan kartu memori dan beberapa bentuk kata sandi atau nomor identifikasi pribadi (PIN). Aplikasi yang umum digunakan adalah mesin anjungan tunai mandiri (ATM). Kartu memori, jika digabungkan dengan PIN atau kata sandi, memberikan keamanan yang jauh lebih besar daripada kata sandi saja. Musuh harus mendapatkan kepemilikan fisik dari kartu (atau dapat menduplikatnya) dan juga harus mendapatkan pengetahuan tentang PIN. Di antara kelemahan potensial adalah sebagai berikut [NIST95]:

- **Membutuhkan pembaca khusus:** Hal ini meningkatkan biaya penggunaan token dan menciptakan persyaratan untuk menjaga keamanan perangkat keras dan perangkat lunak pembaca.
- **Kehilangan token:** Token yang hilang untuk sementara waktu mencegah pemiliknya untuk mendapatkan akses sistem. Dengan demikian, ada biaya administrasi untuk mengganti token yang hilang. Sebagai tambahan, jika token ditemukan, dicuri, atau dipalsukan, maka pihak yang tidak bertanggung jawab hanya perlu menentukan PIN untuk mendapatkan akses yang tidak sah.
- **Ketidakpuasan pengguna:** Walaupun pengguna mungkin tidak mengalami kesulitan dalam menerima penggunaan kartu memori untuk akses ATM, namun penggunaannya untuk akses komputer mungkin dianggap tidak nyaman.

Kartu Pintar

Berbagai macam perangkat memenuhi syarat sebagai token pintar. Ini dapat dikategorikan dalam empat dimensi yang tidak saling terpisah:

- **Karakteristik fisik:** Token pintar memiliki mikroprosesor tertanam. Token pintar yang terlihat seperti kartu bank disebut kartu pintar. Token pintar lainnya dapat terlihat seperti kalkulator, kunci, atau benda portabel kecil lainnya.
- **Antarmuka pengguna:** Antarmuka manual termasuk keypad dan tampilan untuk interaksi manusia/token.
- **Antarmuka elektronik:** Kartu pintar atau token lainnya memerlukan antarmuka elektronik untuk berkomunikasi dengan pembaca/penulis yang kompatibel. Sebuah kartu dapat memiliki salah satu atau kedua jenis antarmuka berikut ini:

  — **Kontak:** Kartu pintar kontak harus dimasukkan ke dalam pembaca kartu pintar dengan koneksi langsung ke pelat kontak konduktif pada permukaan kartu (biasanya berlapis emas). Transmisi perintah, data, dan status kartu berlangsung melalui titik kontak fisik ini.

  -Tanpa kontak**:** Kartu nirkontak hanya memerlukan kedekatan dengan pembaca. Baik pembaca maupun kartu memiliki antena, dan keduanya berkomunikasi menggunakan frekuensi radio. Sebagian besar kartu nirkontak juga mendapatkan daya untuk chip internal dari sinyal elektromagnetik ini. Kisarannya biasanya satu setengah hingga tiga inci untuk kartu yang tidak bertenaga baterai, ideal untuk aplikasi seperti pintu masuk gedung dan pembayaran yang membutuhkan antarmuka kartu yang sangat cepat.

- **Protokol otentikasi:** Tujuan dari smart token adalah untuk menyediakan sarana autentikasi pengguna. Kita dapat mengklasifikasikan protokol otentikasi yang digunakan dengan smart token ke dalam tiga kategori:
  - **Statis:** Dengan protokol statis, pengguna mengautentikasi dirinya sendiri ke token dan kemudian token mengautentikasi pengguna ke komputer. Paruh terakhir dari protokol ini mirip dengan pengoperasian token memori.

— **Generator kata sandi dinamis:** Dalam hal ini, token menghasilkan kata sandi unik secara berkala (misalnya, setiap menit). Kata sandi ini kemudian dimasukkan ke dalam sistem komputer untuk otentikasi, baik secara manual oleh pengguna atau secara elektronik melalui token. Token dan sistem komputer harus diinisialisasi dan terus disinkronkan sehingga komputer mengetahui kata sandi yang berlaku untuk token ini.

— **Tantangan-tanggapan:** Dalam hal ini, sistem komputer menghasilkan tantangan, seperti serangkaian angka acak. Token pintar menghasilkan respons berdasarkan tantangan tersebut. Sebagai contoh, kriptografi kunci publik dapat digunakan dan token dapat mengenkripsi string tantangan dengan kunci privat token.

Untuk otentikasi pengguna, kategori token pintar yang paling penting adalah kartu pintar, yang memiliki tampilan seperti kartu kredit, memiliki antarmuka elektronik, dan dapat menggunakan salah satu jenis protokol yang baru saja dijelaskan. Sisa dari bagian ini membahas kartu pintar.

Kartu pintar berisi seluruh mikroprosesor, termasuk prosesor, memori, dan port I/O. Beberapa versi menggabungkan sirkuit pemrosesan bersama khusus untuk operasi kriptografi untuk mempercepat tugas penyandian dan penguraian pesan atau membuat tanda tangan digital untuk memvalidasi informasi yang ditransfer. Pada beberapa kartu, port I/O dapat diakses secara langsung oleh pembaca yang kompatibel dengan menggunakan kontak listrik yang terbuka. Kartu lainnya mengandalkan antena tertanam untuk komunikasi nirkabel dengan pembaca.

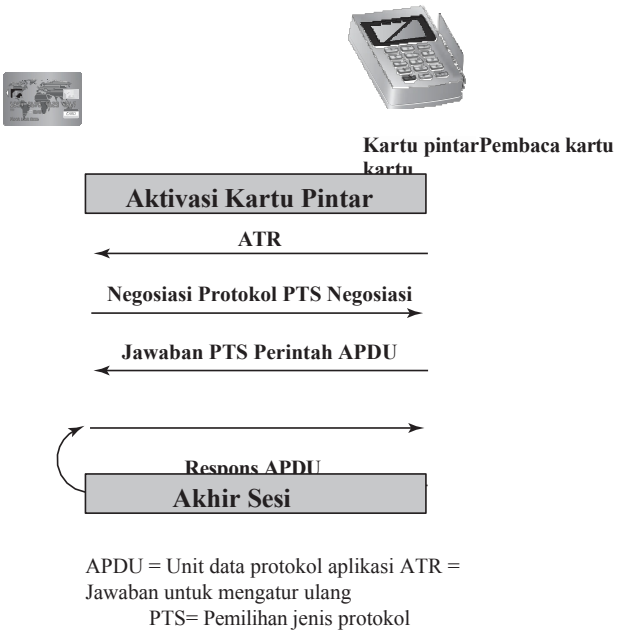
Kartu pintar pada umumnya memiliki tiga jenis memori. Memori hanya-baca (ROM) menyimpan data yang tidak berubah selama masa pakai kartu, seperti nomor kartu dan nama pemegang kartu. ROM yang dapat diprogram yang dapat dihapus secara elektrik (EEPROM) menyimpan data aplikasi dan program, seperti protokol yang dapat dijalankan oleh kartu. EEPROM juga menyimpan data yang dapat berubah seiring waktu. Misalnya, pada kartu telepon, EEPROM menyimpan sisa waktu bicara. Memori akses acak (RAM) menyimpan data sementara yang dihasilkan saat aplikasi dijalankan.

Gambar 3.5 mengilustrasikan interaksi umum antara kartu pintar dan pembaca atau sistem komputer. Setiap kali kartu dimasukkan ke dalam pembaca, reset dimulai oleh pembaca untuk menginisialisasi parameter seperti nilai jam. Setelah fungsi reset dilakukan, kartu merespons dengan pesan Answer to Reset (ATR). Pesan ini mendefinisikan parameter dan protokol yang dapat digunakan oleh kartu dan fungsi yang dapat dijalankan. Terminal mungkin dapat mengubah protokol yang digunakan dan parameter lainnya melalui perintah pemilihan jenis protokol (PTS). Tanggapan PTS kartu akan mengonfirmasi protokol dan parameter yang akan digunakan. Terminal dan kartu sekarang dapat menjalankan protokol untuk melakukan aplikasi yang diinginkan.

Kartu Identitas Elektronik

Aplikasi yang semakin penting adalah penggunaan kartu pintar sebagai kartu identitas nasional untuk warga negara. Kartu identitas elektronik nasional (eID) dapat memiliki fungsi yang sama dengan kartu identitas nasional lainnya, dan kartu serupa seperti surat izin mengemudi, untuk mengakses layanan pemerintah dan komersial. Selain itu, kartu eID dapat memberikan bukti identitas yang lebih kuat dan dapat digunakan dalam berbagai macam aplikasi. Pada dasarnya, kartu eID adalah kartu

pintar yang telah diverifikasi oleh pemerintah nasional sebagai valid dan otentik.



Gambar 3.5 Pertukaran Kartu Cerdas/Pembaca

Salah satu penerapan eID terbaru dan tercanggih adalah kartu eID Jerman, *neuer Personalausweis* [POLL12]. Kartu ini memiliki data yang dapat dibaca manusia yang tercetak di permukaannya, termasuk yang berikut ini:

- **Data pribadi:** Seperti nama, tanggal lahir, dan alamat; ini adalah jenis informasi tercetak yang ditemukan pada paspor dan SIM.
- **Nomor dokumen:** Pengenal unik sembilan karakter alfanumerik untuk setiap kartu.
- **Nomor akses kartu (CAN):** Nomor acak desimal enam digit yang dicetak pada bagian muka kartu. Nomor ini digunakan sebagai kata sandi, seperti yang akan dijelaskan selanjutnya.
- **Zona yang dapat dibaca mesin (MRZ):** Tiga baris teks yang dapat dibaca manusia dan mesin di bagian belakang kartu. Ini juga dapat digunakan sebagai kata sandi.

FUNGSI IDUL FITRI Kartu ini memiliki tiga fungsi elektronik terpisah berikut ini, masing-masing dengan kumpulan data yang dilindungi (Tabel 3.3):

- **ePass:** Fungsi ini diperuntukkan bagi penggunaan pemerintah dan menyimpan repetisi digital identitas pemegang kartu. Fungsi ini mirip dengan, dan dapat digunakan untuk, paspor elektronik. Layanan pemerintah lainnya juga dapat menggunakan ePass. Fungsi ePass harus diimplementasikan pada kartu.
- **eID:** Fungsi ini untuk penggunaan tujuan umum dalam berbagai aplikasi pemerintah dan komersial. Fungsi eID menyimpan catatan identitas yang