Analisis Penggunaan Public Key Cryptography pada Passkey Sebagai Pengganti Password

Samuel Aristides 18219080
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
samuelabukit@gmail.com

***Abstrak*—Password telah lama digunakan sebagai metode verifikasi dan otentikasi identitas pengguna di sebuah *website*. Password pun memiliki berbagai kelemahan dan rentan terhadap serangan. Terdapat metode otentikasi baru yang memanfaatkan kriptografi kunci publik dan baru mulai digunakan, yaitu passkey. Dari analisis yang dilakukan, passkey dapat membantu mengatasi kekurangan yang umumnya dimiliki sebuah password. Tidak ada kriptografi yang aman, dan passkey juga punya kelemahan, namun untuk saat ini, passkey bisa menjadi opsi yang lebih baik dibandingkan password.**

Kata kunci—kriptografi, passkey, kunci publik, password, keamanan siber, login

I. PENDAHULUAN

Metode autentikasi untuk sebuah akun yang paling sering digunakan saat ini adalah dengan menggunakan pasangan email atau *username* dan sebuah kata sandi. Kata sandi telah menjadi teknik autentikasi yang paling umum digunakan. Konsep penggunaan kata sandi untuk masuk ke sebuah akun sudah digunakan sejak awal adanya sistem komputasi, dengan digunakannya konsep ini pada komputer CTSS milik Massachusetts Institute of Technology (MIT) tahun 1961 [1].

Semakin berkembangnya jumlah pengguna teknologi juga berarti semakin banyak akun yang dimiliki seseorang. Sebuah survei menyatakan bahwa sekitar 14% pengguna internet memiliki lebih dari 25 akun, 28% menyatakan memiliki 11 sampai 25 akun, dan 30% berkata mereka memiliki terlalu banyak akun untuk dihitung [1]. Setiap akun membutuhkan sebuah kata sandi, dan praktik terbaiknya adalah satu kata sandi sebaiknya hanya dipakai di satu akun. Namun, dengan banyaknya akun yang dapat dimiliki satu orang, kemungkinan besar praktik tersebut tidak dilakukan. Survei yang sama menunjukkan rata-rata satu kata sandi digunakan untuk lima akun yang berbeda, sementara 71% akun menggunakan kata sandi yang digunakan di lebih dari satu akun [1].

Keamanan sebuah kata sandi sendiri menjadi sebuah masalah besar. Seringkali saat membuat sebuah akun, sistem akan memberi syarat atau saran mengenai karakter yang digunakan untuk kata sandi, seperti angka, huruf besar dan kecil, serta simbol. Namun, seringkali pengguna menggunakan kata sandi yang mudah ditebak atau menggunakan informasi pribadi mereka, seperti tanggal lahir. Buktinya, sebanyak 59% masyarakat di Amerika menggunakan nama atau tanggal lahir mereka di dalam kata sandi yang digunakan.

Kata sandi menjadi salah satu faktor risiko keamanan terbesar dari sebuah perusahaan yang memiliki sistem *login*. Selain risiko yang telah disebutkan sebelumnya, insiden terbobolnya data pengguna yang paling krusial adalah data kata sandi, meski seringkali kata sandi yang terbobol sudah tersimpan dalam bentuk *hash* yang tidak bisa digunakan langsung, namun bisa diretas dengan teknik *brute force* atau mencoba semua kemungkinan kata sandi. Terdapat jenis serangan lainnya, salah satunya *phishing*. *Phishing* adalah sebuah ancaman besar, di mana aktor jahat membuat tampilan *website* yang sangat mirip dengan yang asli, atau seringkali sama persis. Mereka juga menggunakan nama *domain* yang mirip dengan aslinya. Tujuan *phishing* adalah menipu pengguna agar mereka memasukkan informasi akunnya ke *website* palsu tersebut. Pemilik *website* palsu tersebut kemudian bisa mengakses akun pengguna yang tertipu dan mengganti kata sandinya. Sebuah sumber mengatakan 89% organisasi mengalami serangan *phishing* selama tahun 2022 [2].

Kata sandi, meski popularitasnya, memiliki banyak masalah. Salah satu solusi yang saat ini mulai marak digunakan adalah aplikasi pengelola kata sandi, atau yang disebut *Password Manager*. Namun, tidak dapat dipungkiri bahwa membuat kata sandi berarti menyimpan sebuah rahasia bersama yang diketahui oleh pengguna dan sebuah sistem basis data. Teknik autentikasi *multi-factor* (MFA) juga membantu keamanan kata sandi karena menghilangkan adanya *shared secret*, namun menambah kompleksitas *password* dan juga masih memungkinkan terjadinya *phishing*.

Sebuah konsep baru yang dibilang akan membuat pengguna meninggalkan kata sandi adalah teknologi Passkey. Pada dasarnya, Passkey adalah teknologi autentikasi yang menggunakan konsep kriptografi infrastruktur kunci publik. Pengguna menyimpan kunci privat di perangkatnya, dan server hanya menyimpan kunci publiknya saja. Konsep ini baru mulai digunakan oleh segelintir perusahaan, seperti Google yang mulai mengimplementasikan metode ini untuk masuk ke akun Google [3]. Makalah ini bertujuan untuk menganalisis metode autentikasi yang digunakan oleh teknologi Passkey, dan membandingkan keamanannya dengan teknologi kata sandi saat ini.

II. METODOLOGI PENELITIAN

A. *Metode Penelitian*

Data penelitian pada makalah ini didapatkan seluruhnya dari sumber yang dapat diakses di internet. Sumber yang tersedia masih terbatas karena teknologi ini masih sangat baru pada waktu penulisan, dan spesifikasi teknikal Passkey menjadi salah satu sumber utama studi literatur. Penelitian ini juga akan dilanjutkan dengan perbandingan untuk berbagai risiko keamanan yang saat ini dihadapi kata sandi, dan bagaimana Passkey mungkin mengurangi risiko tersebut.

B. *Batasan Penulisan*

Penelitian pada makalah ini dibatasi dengan bahasan algoritma pengamanan yang digunakan oleh Passkey, serta analisis perbandingan keamanan dengan kata sandi.

III. DASAR TEORI

A. *Kriptografi*

Kriptografi adalah sebuah ilmu atau metode yang digunakan untuk melindungi sebuah informasi agar tidak bisa dilihat atau digunakan oleh pihak tidak berwenang. Kriptografi seringkali menggunakan teknik matematis untuk membantu mengamankan data.

Kriptografi memiliki dua fungsi utama yang dilakukan dalam mengamankan data. Kedua proses tersebut adalah enkripsi dan dekripsi. Enkripsi adalah proses mengubah data dari data bermakna ke data terenkripsi yang tidak bisa digunakan langsung. Proses enkripsi dilakukan dengan menggunakan sebuah kunci. Dekripsi adalah proses mengubah data hasil enkripsi menjadi data yang kembali bisa digunakan. Dekripsi juga dilakukan menggunakan sebuah kunci.

Kriptografi dapat dibagi menjadi berbagai macam pengelompokkan. Salah satu pengelompokan tersebut adalah sistem kunci enkripsi dan dekripsi yang dibagi menjadi dua, yaitu kriptografi dengan kunci simetris dan asimetris.

Kriptografi dengan kunci simetris berarti kunci yang sama digunakan untuk melakukan proses enkripsi dan dekripsi. Pada sistem kunci simetris, kedua belah pihak memiliki kunci yang sama persis. Pihak pengirim akan mengekripsi data menjadi data terenkripsi yang tidak bermakna. Pihak penerima yang memiliki kunci yang sama seperti yang digunakan saat enkripsi dapat melakukan dekripsi.

Kriptografi dengan kunci asimetris berarti kunci yang berbeda digunakan untuk melakukan proses enkripsi dan dekripsi. Disebut juga kriptografi kunci publik, kriptografi dengan kunci asimetris memiliki dua kunci, yaitu kunci privat dan kunci publik. Konsep ini dapat digunakan untuk enkripsi dan dekripsi data, dengan enkripsi memakai kunci publik dan dekripsi menggunakan kunci privat, atau dapat digunakan untuk membuat tanda tangan digital. Proses anda tangan digital dilakukan dengan membubuhkan *string* di akhir sebuah *file*. *String* untuk tanda tangan digital didapatkan dengan melakukan *hash* kunci privat terhadap *file*.

Kriptografi perlu memenuhi empat aspek utama, yaitu *confidentiality*, *integrity*, *authentication*, dan *non-repudiation*. *Confidentiality* berarti data tidak terlihat oleh pihak tidak berwenang. *Integrity* berarti data tidak dapat dimanipulasi atau dihancurkan. *Authentication* berarti semua pihak dapat memverifkasi identitas satu sama lain. *Non-repudiation* berarti suatu pihak tidak dapat menyangkal telah mengenkripsi atau mengirim suatu data.

B. *Password*

Password atau kata sandi adalah sebuah rangkaian karakter yang umumnya digunakan untuk melakukan autentikasi dan memverifikasi identitas pengguna. Tujuan penggunaan *password* adalah untuk melindungi akses oleh pihak tidak berwenang.

Sebuah *password* yang baik biasanya menggunakan kombinasi dari huruf besar, huruf kecil, angka, dan simbol. Disarankan juga untuk tidak menggunakan informasi pribadi di dalam sebuah *password*, seperti tanggal lahir atau nama diri sendiri. Hal-hal ini bisa melemahkan *password* karena semakin mudah ditebak. *Password* juga semakin aman apabila menggunakan karakter acak yang tidak berarti, meskipun akan menyulitkan pengguna untuk mengingatnya.

Pada umumnya, *password* disimpan di sebuah basis data dalam bentuk yang sudah melewati proses *hashing* dengan algoritma tertentu. Pada saat pengguna ingin mengakses suatu akun, pengguna akan diminta untuk memasukkan *password*. Sistem akan melakukan proses *hashing* dari masukan yang diberikan pengguna, lalu mencocokannya dengan data yang tersimpan di basis data.

Terdapat beberapa teknologi yang membantu keamanan *password*, seperti program Password Manager. Password Manager adalah sebuah program yang dapat digunakan untuk menyimpan *password* pengguna. Seringkali, program ini juga memiliki kemampuan membuat *password* dengan karakter acak dan sesuai dengan teknik-teknik terbaik, seperti menyertakan angka dan simbol, serta mengatur berapa panjang *password*. Password Manager akan menyimpan *password* acak ini, sehingga pengguna tidak perlu mengingatnya. Pengguna hanya perlu memiliki satu *password* untuk Password Manager agar dapat mengakses semua *password* yang dimiliki pengguna.

Teknologi lain yang membantu keamanan *password* adalah metode autentikasi dua faktor, atau autentikasi multi-faktor. *Password* menggunakan konsep *shared secret*, yang artinya terdapat sebuah rahasia yang diketahui bersama antara dua pihak, meski server mengetahui rahasia tersebut dalam bentuk lain. Autentikasi dua faktor berfungsi sebagai faktor yang bukan merupakan *shared secret*. Autentikasi dua faktor pada umumnya menambah lapisan keamanan, seperti verifikasi menggunakan PIN, informasi biometrik, atau dengan sistem TOTP (*Time-based One Time Password)*.

Seperti semua teknik keamanan, *password* pun memiliki kelemahannya sendiri. Beberapa di antaranya adalah *phishing*, *brute force attack*, *keylogging* (merekam ketikan *keyboard* pengguna saat memasukkan *password*), kebocoran data, dan lain-lain.

C. *Public Key Infrastructure*

Public Key Infrastructure adalah seluruh infrastruktur yang mengatur penggunaan kunci publik dalam lingkungan digital. Infrastruktur ini termasuk kebijakan, otoritas, prosedur, dan teknologi yang digunakan.

Penggunaan utama Public Key Infrastructure pada saat ini adalah untuk membuat koneksi aman dari klien atau pengguna ke sebuah *website.* Infrastruktur ini memastikan bahwa situs yang dikunjungi aman dan telah terverifikasi sebagai *website* asli. Terdapat sertifikat digital yang dibuat oleh Certification Authority (CA) atau otoritas sertifikasi yang membuktikkan hal tersebut.

CA menyimpan kunci pada sebuah repositori kunci yang dapat diakses oleh semua pihak. Untuk menunjukkan keaslian sebuah sertifikat, sertifikat telah ditandatangani menggunakan kunci privat milik CA. Pihak luar dapat memeriksa keaslian suatu sertifikat digital memakai kunci publik milik CA.

D. *Passkey*

Gambar 1 Logo Passkey (fidoalliance.org)

Passkey adalah sebuah konsep baru yang dibuat oleh organisasi FIDO yang terdiri dari beberapa nama besar dalam dunia teknologi, seperti Google, Microsoft, dan Apple. Aliansi ini dibentuk dengan tujuan mengurangi penggunaan *password*.

Aliansi ini telah membuat beberapa spesifikasi standar yang dapat diimplementasikan oleh organisasi lain untuk membuat sistem autentikasi akun yang tidak menggunakan *password*. Standar utama yang digunakan adalah standar FIDO2, yaitu sebuah standar yang menggunakan standar W3C yang Bernama WebAuthn. WebAuthn memungkinkan pengguna untuk melakukan login tanpa *password* dengan menggunakan metode verifikasi lain seperti informasi biometric pengguna dan PIN. Rincian lebih lengkap dari Passkey akan dijelaskan pada bab berikutnya.

E. *Digital Signature*

Tanda tangan digital, atau *digital signature*, adalah sebuah mekanisme dalam kriptografi yang memungkinkan suatu pihak memverifikasi pihak lain. Sama seperti sebuah tanda tangan fisik, tanda tangan digital biasanya dibubuhkan di suatu bagian dokumen, dan menjadi identitas dari pembuat dokumen tersebut. Sama juga seperti tanda tangan fisik, tanda tangan digital bisa diperiksa keasliannya.

Tanda tangan digital menggunakan salah satu konsep yang telah dijelaskan sebelumnya, yaitu kunci publik dan kunci privat. Kunci privat dimiliki oleh pihak yang memiliki dokumen asli, sementara kunci publik dapat diakses oleh siapa saja untuk memastikan identitas pemilik kunci privat.

Pada proses tanda tangan dokumen, pemilik kunci pribadi akan melakukan proses kriptografi pada dokumen yang ingin ditandatangani. Pada umumnya, proses yang digunakan adalah proses *hashing* dengan algoritma tertentu untuk data yang ingin ditandatangani, kemudian mengenkripsi hasil *hash* tersebut menggunakan kunci pribadi. Hasil enkripsi tersebut menghasilkan sebuah rangkaian karakter yang dapat dibubuhkan di dokumen.

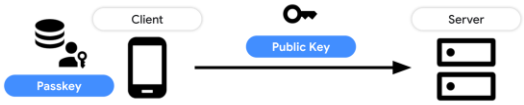
Untuk memverifikasi tanda tangan dokumen, pihak yang ingin melakukan verifikasi menggunakan kunci publik milik penandatangan data. Tanda tangan yang telah dibubuhkan perlu didekripsi menggunakan kunci publik. Hal ini akan menghasilkan *hash* dari data. Kemudian, data perlu di-*hash* menggunakan algoritma yang sama seperti saat proses pembuatan tanda tangan. Verifikasi tanda tangan dilakukan dengan membandingkan hasil *hash* yang didapatkan dari dekripsi dan *hashing* data. Apabila kedua hasil *hash* sama, maka data belum diubah dan identitas terverifikasi.

IV. PEMBAHASAN

Pada bab ini, akan dibahas mengenai metode registrasi dan otentikasi yang dimodelkan oleh Passkey. Bab ini juga akan memberikan penjelasan mengenai serangan dan kelemahan yang umumnya dihadapi oleh *password*, dan akan menjelaskan bagaimana konsep Passkey dapat mengatasi kelemahan tersebut.

A. *Metode Registrasi Passkey*

Pada dasarnya, konsep registrasi Passkey adalah dengan menggunakan infrastruktur kunci publik. Secara garis besar, teknis metode registrasi Passkey adalah pengguna yang ingin membuat akun akan diberikan opsi untuk membuat sebuah Passkey dibanding akun biasa dengan sebuah *username* dan *password.* Apabila pengguna setuju, maka sistem operasi perangkat pengguna akan menggunakan metode verifikasi yang ada, seperti PIN atau informasi biometrik seperti sidik jari. Setelah itu, aplikasi otentikasi di perangkat pengguna akan membuat sebuah pasangan kunci privat dan kunci publik. Kunci publik diberikan ke server untuk disimpan, sementara kunci privat disimpan di perangkat pengguna.



Gambar 2 Passkey menurut Google (developers.google.com)

Secara rinci, terdapat beberapa implementasi yang dilakukan. Google sendiri menyatakan beberapa komponen yang diperlukan untuk melakukan registrasi [4]. Komponen tersebut adalah:

- RP ID (Relying Party ID): Relying Party adalah perusahaan atau *website* tempat pengguna akan membuat akun. RP ID adalah tanda pengenal untuk *website* tersebut yang bisa berupa nama domain *website*.
- Informasi pengguna: *Username* pengguna beserta nama dan nomor ID pengguna

- Credentials to exclude: informasi mengenai Passkey yang sudah ada sebelumnya untuk mencegah registrasi yang duplikat
- Passkey types: Jenis Passkey yang digunakan untuk otentikasi, bisa menggunakan *platform authenticator* atau *roaming authenticator*. *Platform authenticator* adalah otentikasi yang menggunakan fitur keamanan milik perangkat, seperti PIN atau informasi biometrik. *Roaming authenticator* adalah otentikasi yang menggunakan perangkat terpisah, seperti piranti keras USB khusus untuk *login* (USB Security Keys).

Relying Party dapat menawarkan pengguna opsi membuat Passkey. Apabila pengguna setuju, pengguna dapat memverifikasi persetujuan itu dengan membuka kunci yang biasanya digunakan untuk membuka perangkat. Setelah itu, perangkat otomatis membuat sebuah Passkey baru, dan informasi kunci publik diberikan kepada server dari Relying Party. Informasi diberikan beserta dengan ID dari informasi tersebut, agar dapat diakses untuk *login* kembali.

Implementasi Google tersebut dibuat berdasarkan standar yang telah dibuat FIDO. FIDO sendiri membuat standar mereka berdasarkan standar milik W3C (World Wide Web Consortium) yang juga dibuat dengan kolaborasi FIDO, yaitu WebAuthn. Menurut dokumentasinya, terdapat lebih dari 20 langkah untuk melakukan registrasi [5]. Berikut adalah uraian singkat dari langkah-langkah tersebut.

Pada saat passkey akan dibuat, sistem akan mengatur beberapa pengaturan terlebih dahulu. Pengaturan tersebut terdiri dari identitas dari Relying Party serta pengguna berupa ID masing-masing pihak. Pengaturan ini juga membutuhkan sebuah rangkaian karakter acak yang akan disebut sebagai *challenge* atau tantangan untuk registrasi ini. Selain itu, algoritma perlu ditentukan pada pengaturan ini. Algoritma mengacu pada laman IANA untuk tanda tangan dan enkripsi [6]. Bagian penting lainnya dari pengaturan ini adalah menentukan pengaturan pengesahan yang akan digunakan Relying Party untuk memverifikasi pengesahan dari otentikasi perangkat.

Pada saat ini, *website* sudah akan meminta izin pengguna untuk membuat passkey. Apabila pengguna setuju, perangkat akan menerima pengaturan di atas, dan membuat sebuah kunci publik berdasarkan pengaturan yang diberikan sebelumnya. Perangkat juga mengumpulkan data informasi pengguna yang ingin melakukan registrasi.

Selanjutnya, *website* akan menerima respons dari perangkat berupa data pengguna dan kunci publik. Data pengguna diverifikasi berdasarkan tipe perintah, yaitu melakukan registrasi. *Challenge* dari data pengguna juga diverifikasi dan harus sama dengan yang diberikan pada pengaturan. Data asal Relying Party juga diverifikasi dan harus sama, beserta status koneksi aman yang telah terbentuk. Data pengguna kemudian di-*hash* dengan algoritma SHA-256.

Selanjutnya, bagian yang diperiksa adalah bagian pengesahan yang berasal dari sistem otentikasi perangkat. Bagian pengesahan didekripsi terlebih dahulu, lalu beberapa data dicocokkan. Data pertama adalah hasil *hash* ID dari Relying Party serta hasil *hash* ID Relying Party yang sudah ada di pengesahan. Algoritma yang digunakan adalah SHA-256. Data berikutnya yang diverifikasi adalah memastikan algoritma yang dipakai untuk kunci publik ada pada salah satu algoritma yang diperbolehkan di pengaturan. Data berikutnya adalah pernyataan pengesahan yang perlu diverifikasi kebenarannya dengan melihat tanda tangan pernyataan pengesahan. Tanda tangan dibuat dengan menggabungkan data otentikator dari perangkat dengan hasil *hashing* data pengguna, lalu menandatanganinya dengan kunci privat perangkat untuk pengesahan. Validasi juga bisa dilihat dengan cara mencari ID otentikator yang ada di bagian pengesahan dengan *database* otentikator terpercaya, atau menggunakan sertifikat digital yang mungkin ada.

Apabila seluruh tahap diatas berhasil dilewati, sistem akan mendaftarkan informasi baru tersebut dengan informasi akun yang telah diberikan di bagian pengaturan. *Website* perlu menghubungkan akun pengguna dengan ID informasi serta kunci publik yang berhubungan.

B. Metode Otentikasi Passkey

Pada dasarnya, otentikasi passkey hanya sebatas verifikasi tanda tangan yang dilakukan di atas data yang sangat serupa dengan data pengesahan, yaitu hasil gabungan antara data otentikator perangkat serta data pengguna. Tanda tangan tersebut akan didekripsi dengan kunci publik yang tersimpan, lalu dibandingkan dengan hasil menandatangani data yang ada. Menurut standar FIDO, terdapat lebih dari 20 langkah dalam melakukan otentikasi. Berikut adalah uraian singkat dari standar tersebut.

Sama seperti saat registrasi, sistem akan memberikan berbagai pengaturan ke otentikator di perangkat. Pengaturan tersebut wajib berisi sebuah *challenge* yang akan ditandatangani oleh otentikator. Terdapat beberapa pengaturan opsional lainnya, seperti ID Relying Party.

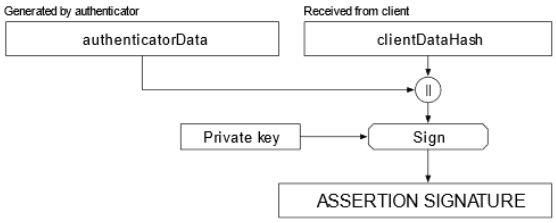
Pada tahap ini, pengguna akan dihadapkan dengan pilihan untuk melakukan *login* dengan menggunakan passkey. Apabila pengguna menyetujui *login* ini, pengguna akan diberikan opsi untuk melakukan verifikasi identitas diri. Setelah terverifikasi, otentikator di perangkat akan menandatangani data yang terdiri dari data pengguna, data otentikator yang berisi pengaturan dari *website*, dan tanda tangan itu sendiri.

Tahap penting berikutnya adalah mengidentifikasi pengguna yang melakukan permintaan *login* dan memverifikasi identitas tersebut. Verifikasi dapat dilakukan dengan menggunakan ID identitas yang telah diregistrasi sebelumnya. Kunci publik yang cocok dengan ID tersebut akan digunakan untuk verifikasi.

Selanjutnya, *website* akan menerima data pengguna, data otentikasi, dan tanda tangan. *Website* akan memeriksa bahwa tipe permintaan dari pengguna adalah untuk melakukan *login*. *Website* juga perlu memeriksa bahwa *challenge* yang telah ditetapkan di pengaturan sama dengan yang diterima di data ini. *Website* juga perlu memastikan bahwa data asal Relying Party sama dengan *website*, beserta status koneksi aman yang telah terbentuk. Data pengguna kemudian di-*hash* dengan algoritma SHA-256.

Selanjutnya, sistem perlu memeriksa hasil *hashing* dari ID Relying Party pada data otentikator, sesuai dengan hasil *hashing* dari ID Relying Party dengan algoritma SHA-256.

Bagian utama dari verifikasi ini adalah di tahap berikut, yaitu memverifikasi bahwa tanda tangan yang diterima dari otentikator adalah tanda tangan yang benar. Hal ini bisa dikonfirmasi dengan tanda tangan publik yang dimiliki oleh *website*. Tanda tangan diperoleh dengan menggabungkan data otentikator dan hasil *hashing* dari data pengguna. Apabila tanda tangan terverifikasi, maka pengguna dapat *login*. Proses ini dapat digambarkan lebih jelas dengan grafik berikut.



Gambar 3 Proses Verifikasi Pengguna (w3.org)

C. Melawan Serangan Phishing

Phishing adalah tindakan penipuan yang umumnya bertujuan untuk mendapatkan informasi atau identitas seseorang. Informasi tersebut umumnya akan digunakan untuk melakukan pembajakan akun. *Phishing* dapat terjadi dengan menggunakan *website* palsu dan *domain* palsu, namun juga dapat terjadi melalui email dan bahkan telepon apabila memanfaatkan konsep *social engineering*.

Pada umumnya, tahapan yang dilakukan dalam *phishing* adalah sebagai berikut:

- Aktor jahat membuat sebuah pesan yang menyesatkan atau mengatasnamakan pihak lain. Tujuan tahap ini adalah untuk mendapatkan perhatian dari pihak yang ingin ditipu. Beberapa taktik yang sering ditemukan adalah membuat seseorang panik atau penasaran.
- Aktor jahat sudah menyiapkan *website* dengan tampilan yang sama persis seperti tampilan *website* yang ingin dicuri identitasnya. Selain itu, umumnya telah disiapkan juga *domain* yang sangat mirip dengan *website* aslinya, misalnya hanya salah satu huruf. Hal ini bertujuan untuk menjaring orang-orang yang salah mengetik nama situs.
- Saat pengguna sudah masuk ke *website* palsu, mereka yang tertipu umumnya akan memasukkan informasi akun ke *website* palsu. Namun, *website* palsu tersebut tidak terhubung ke *website* asli, dan data yang dimasukkan dapat dilihat oleh aktor jahat. Dari sini, aktor jahat dapat memasukkan informasi akun yang baru mereka dapatkan untuk dimasukkan di *website* yang asli. Setelah itu, mereka mendapatkan kontrol penuh atas akun.

Secara logika, sebuah kata sandi tidak diketahui oleh orang lain selain pemilik akun. Aktor jahat pun mengetahui hal itu, maka satu cara mereka dapat memperoleh kata sandi tersebut adalah dengan mendapatkannya langsung dari pemiliknya.

Password memiliki kelemahan di mana pengguna harus memasukkan *password* tersebut ke sebuah halaman. Karena itu, terdapat sebuah rahasia, dalam hal ini *password*, yang diketahui bersama antara pengguna dan sistem *website*. Hal ini menjadi titik lemah, karena apabila masukan tersebut bisa diketahui oleh aktor jahat, maka mereka pun akan mengetahui rahasia bersama tersebut.

Passkey tidak memiliki risiko serangan *phishing*. Hal ini dimungkinkan karena dua hal, yaitu karena algoritma otentikasi yang dimiliki passkey, serta bantuan sistem otentikasi dari perangkat. Algoritma otentikasi passkey meminta penggunanya untuk melakukan verifikasi dengan memberikan sebuah *challenge* dan meminta perangkat untuk menandatanganinya. Karena pengguna tidak memasukkan apapun ke halaman *website*, tidak ada informasi yang bisa dicuri dan digunakan kembali oleh aktor jahat, berbeda dengan *password*.

Faktor kedua adalah sistem otentikator yang ada di perangkat pengguna. Pada dasarnya, otentikator di perangkat pengguna juga akan membantu memeriksa *domain* halaman yang diakses, seperti melihat ID Relying Party dan mencari ID identitas yang cocok dengan ID Relying Party tersebut. Sistem otentikasi tidak akan memberikan identitas yang valid apabila *domain* yang dikunjungi adalah *domain* palsu.

Sistem ini sebenarnya juga ditemukan pada Password Manager, atau aplikasi pengelola *password*. Apabila pengguna mengakses suatu *website* yang memiliki *domain* berbeda, Password Manager tidak akan memberi *password* untuk *website* tersebut.

D. Melawan Kebocoran Data dan Brute Force Attack

Banyak perusahaan yang pernah mengalami kebocoran data pengguna. Kebocoran data ini mungkin terjadi karena kelalaian pegawai, maupun serangan siber yang komprehensif. Karena algoritma *password*, umumnya sistem *website* akan menyimpan *password* pengguna dalam bentuk lain. Bentuk lain itu merupakan hasil dari berbagai algoritma enkripsi, seperti *hashing*.

Apabila aktor jahat mendapatkan *hash* dari *password* pengguna, sekilas informasi ini tidak berguna. Namun, terdapat sebuah teknik yang disebut *brute force attack*, atau serangan yang memaksa berbagai kombinasi *password* yang mungkin. Serangan lain yang serupa adalah *dictionary attack*, dimana aktor jahat menggunakan sebuah daftar dan mencoba mencocokkan *password*, seperti daftar kata, daftar *password* yang pernah dibobol, atau daftar kata dengan subset huruf atau bahasa tertentu.

Passkey dapat mencegah serangan serupa karena algoritma yang dipakai tidak seperti *password* yang memiliki suatu *hash* yang merujuk langsung ke *password*, meski kadang *hash* dari *password* yang sama bisa berbeda karena teknik *salting*. Passkey menggunakan algoritma kunci publik, yang berarti, meski terjadi kebocoran data kunci publik, aktor jahat tidak dapat menggunakan informasi tersebut tanpa kunci privat pengguna. Kunci privat sendiri hanya ada di perangkat pengguna.

Di masa mendatang, ada kemungkinan bahwa pengguna akan dapat melakukan sinkronisasi passkey antar perangkat, yang berarti akan ada pertukaran kunci privat lewat jaringan internet. Dalam kasus ini, sinkronisasi perlu dilakukan dengan

menggunakan konsep *end-to-end encryption*, yang berarti data dienkripsi sejak sebelum pengiriman hingga sampai di server dan selesai disinkronisasi. Pada kejadian data tersebut bocor, aktor jahat juga akan mengalami kesulitan karena perlu menebak kunci yang digunakan pada *end-to-end encryption*, yang kemungkinan akan dibangkitkan oleh sistem otentikator agar menjadi rumit dan sulit dipecahkan.

Tentu saja, tidak ada kriptografi yang sempurna, dan passkey pun memiliki kelemahan lain yang serupa dengan menebak semua kemungkinan *password*. Hal yang bisa ditebak adalah mencari kombinasi faktor kunci privat dan kunci publik. Namun, pada umumnya operasi seperti ini akan memakan waktu yang sangat lama.

KESIMPULAN

Dari analisis yang telah dilakukan, dapat dilihat bagaimana passkey menggunakan konsep kriptografi yang sudah ada sejak lama untuk melakukan sesuatu yang sudah menjadi kebiasaan sehari-hari, yaitu untuk melakukan *login* dengan menggantikan *password*. Passkey memakai algoritma kunci publik dengan memanfaatkan juga tanda tangan digital untuk memverifikasi pengguna.

Passkey juga dapat menjadi opsi verifikasi yang lebih aman dibandingkan *password*, dilihat dari bagaimana passkey tidak terlalu terpengaruh terhadap serangan yang biasanya dialami *password*. Meski begitu, tidak ada kriptografi yang sempurna, dan algoritma kunci publik juga memiliki kelemahannya sendiri. Seiring berkembangnya dan meluasnya pemakaian metode ini, barulah kita dapat melihat keamanannya dan kelemahannya.

PENGHARGAAN

Penulis mengucapkan syukur kepada Tuhan yang Maha Esa atas anugerah kesehatan dan berkat-Nya sehingga penulis dapat mengerjakan tugas makalah ini. Penulis juga ingin mengucapkan terima kasih kepada teman-teman dan keluarga penulis atas dukungan yang diterima hingga saat ini.

Penulis juga ingin mengucapkan banyak terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T atas pengajaran yang diberikan dalam satu semester keberjalanan mata kuliah II4031 Kriptografi dan Koding. Ilmu yang telah didapatkan menambah wawasan penulis, dan semoga dapat dimanfaatkan lebih lanjut di aktivitas lainnya.

REFERENSI

Berikut adalah referensi yang digunakan untuk membantu pembuatan makalah ini.

[1] https://dataprot.net/statistics/password-statistics/, diakses 20 Mei 2023

[2] https://fidoalliance.org/passkeys/, diakses 20 Mei 2023.

[3] https://support.google.com/accounts/answer/13548313?hl=en, diakses 20 Mei 2023.

[4] https://developers.google.com/identity/passkeys/developer-guide, diakses 21 Mei 2023

[5] https://www.w3.org/TR/webauthn-2/#sctn-registering-a-new-credential, diakses 21 Mei 2023

[6] https://www.iana.org/assignments/cose/cose.xhtml#algorithms, diakses 21 Mei 2023

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Samuel Aristides
18219080