Engineer

Kryptering

att göra data oläsliga utan nyckeln

Vad det är

Kryptering, som en del publikationer från NIST kallar chiffrering, gör läsbara data, klartexten, till chiffertext genom att köra dem genom en publicerad algoritm tillsammans med ett hemligt värde som kallas nyckel. Dekryptering med rätt nyckel ger tillbaka klartexten byte för byte. Algoritmen är offentlig med avsikt: AES, det chiffer som de flesta system väljer, vann en öppen tävling hos NIST år 2000 och är publicerat som en fri standard. Bara nyckeln är hemlig.

Chiffer finns i två familjer. Ett symmetriskt chiffer använder samma nyckel åt båda hållen. AES, specificerat i FIPS 197, krypterar data i block om 128 bitar med en nyckel på 128, 192 eller 256 bitar. ChaCha20, från RFC 8439, är ett strömchiffer som är ungefär tre gånger så snabbt som AES på hårdvara som saknar särskilda instruktioner för AES. Symmetriska chiffer är snabba nog för en hel disk, men båda ändarna behöver samma nyckel, och att få den dit på ett säkert sätt är det svåra.

Kryptografi med publika nycklar löser det med ett nyckelpar, där den ena nyckeln publiceras och den andra hålls privat. Två parter kan var för sig kombinera sin egen privata nyckel med den andras publika nyckel och komma fram till samma delade hemlighet utan att någonsin skicka den. Operationerna med publika nycklar är långsamma, så verkliga system använder båda: ett nyckelutbyte med publika nycklar kommer överens om en ny symmetrisk nyckel, och det symmetriska chiffret bär sedan datan. TLS, protokollet bakom varje adress som börjar med `https://`, fungerar så.

Den moderna idé som betyder mest är autentisering. Ett naket chiffer döljer innehållet men lämnar det öppet för ändringar: i räknarläge vänder en ändrad bit i chiffertexten samma bit i klartexten, och mottagaren dekrypterar det ändrade meddelandet som om det vore äkta. Autentiserad kryptering med associerade

data, AEAD, lägger till en tagg som mottagaren kontrollerar innan den litar på en enda byte. AES-GCM och ChaCha20-Poly1305 är det vanliga paret. TLS 1.3, publicerat som RFC 8446 år 2018, behöll bara AEAD-chiffer och gjorde varje nyckelutbyte framåtsäkert, på engelska forward secret, så att en servernyckel som stjäls senare inte öppnar någon inspelad session.

Använd begreppet i ett beslut

Använd ”Vad är kryptering?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Konton, lösenord och behörigheter ordnade så att alla har exakt den åtkomst deras arbete kräver, och de som slutar förlorar den samma dag.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Skrev 1 000 RBAC-regler som säkrar Linux-servrar — 1,000 åtkomstregler, skrivna för att täppa till en säkerhetslucka

Skrev 1 000 RBAC-regler som säkrar Linux-servrar Hör av er →

Varför det spelar roll

Data stannar sällan på hårdvara som ägaren kontrollerar. De passerar andras nätverk, ligger i någon annans datacenter och lämnar byggnaden på bärbara datorer. Kryptering är det som låter dem passera de platserna säkert: en stulen disk, en avlyssnad kabel eller en läckt lagringsbucket ger chiffertext, och chiffertext utan nyckeln är brus.

Det flyttar problemet snarare än tar bort det. När datan väl är krypterad är dess säkerhet lika med nyckelns säkerhet. Rekommendationen från NIST om nyckelhantering, SP 800-57, jämför nycklar med kombinationen till ett kassaskåp: en angripare som kan kombinationen går rakt förbi det starkaste skåpet, och dålig nyckelhantering kan lätt underminera en stark algoritm. Det mesta av det verkliga arbetet är därför nyckelarbete: att förvara nycklar åtskilda från datan, begränsa hur mycket varje nyckel skyddar, byta dem enligt ett schema och vakta reservkopian lika noga som originalet.

Tillsynsmyndigheterna drar samma gräns. Dataskyddsförordningen, GDPR, räknar pseudonymisering och kryptering av personuppgifter till de säkerhetsåtgärder som nämns i artikel 32. Artikel 34 låter en personuppgiftsansvarig avstå från att underrätta varje berörd person om en personuppgiftsincident när de läckta uppgifterna var obegripliga för den som inte var behörig att se dem, och kryptering är det exempel som förordningen ger.

Kryptering används på två ställen. Kryptering under överföring, oftast TLS, skyddar data på ledningen. Kryptering i vila skyddar dem i lagringen. Själva användningsögonblicket ligger utanför båda: en server som dekrypterar en post för att arbeta med den håller klartexten och nyckeln på en gång, så den som kontrollerar servern läser det som servern läser.

Så ser det ut i praktiken

Under överföring är det dagliga arbetet att hålla certifikaten giltiga. När hundra Docker-applikationer behövde SSL/TLS-certifikat skapade ett ACME-baserat arbetsflöde varje certifikat, förnyade vart och ett innan det gick ut och driftsatte det på värdar med Ubuntu som körde Apache och Nginx. Målet var att ta bort människan ur kedjan, eftersom det är människan som glömmer, och ett utgånget certifikat tar ner en app lika säkert som ett knäckt chiffer.

För lagrade data är den användbara frågan vem som har nyckeln. Företagets implementering av krypterad säkerhetskopiering utanför värddatorn utformades för att utöka återställningsberedskapen bortom en ögonblicksbild hos leverantören. Den använder restic, som krypterar det som lagras med AES-256 i räknarläge och autentiserar det med Poly1305-AES, och dess skript stöder överföring av ögonblicksbilder via SFTP till lagring hos en annan leverantör. I den designen får leverantören chiffertext, och företaget har lösenfrasen i en skyddad fil. Det beskriver de implementerade kontrollerna, inte att de har aktiverats i drift: beskrivningen daterad 2026-10-27 anger att timerenheterna var inaktiverade utan ett konfigurerat repo och redovisar ingen genomförd återställning i drift. Kryptering är en egenskap hos den föreslagna lagringsvägen; för att bevisa att säkerhetskopian kan återställas måste det separata återställningstestet köras.

Till och med vägen som lösenfrasen tar spelade roll. Den läses från en fil, eftersom tjänstehanteraren tolkar escape-tecken i miljövärden, och en lösenfras med ett omvänt snedstreck hade i tysthet blivit en annan än den som skapade repot. Det implementerade månatliga återställningstestet är utformat för att återställa en fil med känt innehåll från repot och jämföra den med det förväntade resultatet. Eftersom testet kräver lösenfrasen skulle en lyckad körning också bekräfta att nyckeln fortfarande öppnar datan.

När plattformen skulle få en andra värd avgjorde samma resonemang hur varje operativ hemlighet delades upp per värd. En sträng för ett repo med säkerhetskopior anger en plats. Separata restic-repon kan ha olika huvudnycklar

även när de använder samma lösenfras. Den som får tag i lösenfrasen och kan läsa båda reponas krypterade data och nyckelfiler kan låsa upp båda, så en delad lösenfras kopplar samman deras exponering. Varje hemlighet blev en mappning med värdnamnet som nyckel, med argumentet nedskrivet bredvid konfigurationen. SP 800-57 för samma resonemang i allmän form: att begränsa vad en nyckel skyddar begränsar skadan när den läcker.

Var det går fel

Nyckeln finns på ett enda ställe. Ett starkt chiffer behandlar en ägare som har glömt lösenordet precis som en angripare. Dokumentationen för restic säger det rakt ut: den som tappar bort sitt lösenord har förlorat sina data för gott. Varje krypterad lagring behöver en andra kopia av sin nyckel, förvarad där det som förstör den första inte når.

Nyckeln ligger bredvid datan. En databas som är krypterad med en nyckel i samma servers konfiguration, eller en säkerhetskopia vars lösenfras ligger i samma bucket, stoppar bara en tjuv som tar halva filerna. Diskkryptering är den tystare varianten: den skyddar en disk som tas ur en avstängd maskin, medan en körande server redan har låst upp sin disk, och en angripare på den servern läser klartext.

Sekretess förväxlas med integritet. Ett system byggt på ett naket chifferläge döljer innehållet men låter ändå en angripare ändra det. Använd en AEAD-konstruktion och kasta varje meddelande vars tagg inte stämmer innan du agerar på en enda byte av det.

En nonce upprepas. AES-GCM och ChaCha20-Poly1305 tar emot en nonce, ett värde som måste vara unikt för varje meddelande under en och samma nyckel. RFC 8439 kallar det den viktigaste säkerhetsaspekten: två meddelanden med samma nonce delar nyckelström och engångsnyckel för autentisering, vilket avslöjar XOR av de två klartexterna. En räknare är ett säkert sätt att skapa nonce-värden, så länge en enda räknare äger nyckeln. Två processer som delar en nyckel och var och en räknar från noll skapar exakt den upprepning som RFC:n varnar för.

Ett lösenord får stå i stället för en nyckel. Människor väljer lösenord; chiffer behöver slumpmässiga nycklar av fast längd. En lösenfras går först genom en nyckelhärledningsfunktion, som restic gör med scrypt, som RFC 7914 utformar för att kräva mycket minne så att gissning i stor skala på specialbyggd hårdvara

förblir dyr. Lagrade inloggningslösenord är ett separat fall: de hör hemma i en sådan funktion som hashvärden, eftersom en server som kan dekryptera dem också kan läcka dem.

Metadatan syns fortfarande. Kryptering döljer innehållet, medan storlek, tidpunkter och adresserna i båda ändar förblir synliga. TLS 1.3 låter en avsändare fylla ut poster för att dölja deras storlek, men utfyllnaden är frivillig, så längden säger ofta mer än ägaren tror.

Läs mer

- NIST Glossary: encryption — varje definition som publikationer från NIST använder, var och en med sitt källdokument.
- FIPS 197: Advanced Encryption Standard (AES) — specifikationen av AES, med blockstorleken och de tre nyckellängderna.
- NIST SP 800-57 Part 1 Rev. 5: Recommendation for Key Management — hur nycklar skapas, förvaras, används och pensioneras, och vad som följer när en läcker.
- RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3 — protokollet bakom `https://`, med poster som bara använder AEAD och framåtsäkert nyckelutbyte.
- RFC 8439: ChaCha20 and Poly1305 for IETF Protocols — ett strömchiffer, en autentiserare och deras gemensamma AEAD-konstruktion, med reglerna för nonce.
- restic: References — hur ett verkligt verktyg för säkerhetskopiering härleder nycklar ur ett lösenord och sedan krypterar och autentiserar sina data.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är kryptering?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/encryption/