Secure System Design: Corporate Encryption

Secure System Design

- Let’s design a secure system
- (Actually, this is the second of three)
- Today: corporate encryption

The Setting

- A large organization (with some comments on smaller ones)
- Multiple locations in several countries

The Usual Questions

- What are the assets?
- What are the threats?
- For encryption, it’s more complicated than usual

Threats that Encryption Addresses

- Theft of stored data
- Theft of data in active use
- Theft from databases
- Eavesdropping
- Authentication

Threats from Using Encryption

- Loss of keys
- Unavailability due to expired keys
- Theft of keys
- Legal issues

Theft of Stored Data

- Many organizations have a lot of archival stored data
- This may include off-site backup media that aren't under the company's physical control
- Should it be encrypted?

(Why Under Someone Else’s Physical Control?)

- Secure offsite backup
- Protect from: theft, fire, other forms of destruction, loss
- There are companies that provide such services

Theft of Actively Used Data

- Actively used data is harder to encrypt
- The keys have to be available to the applications
- Also: where does the decryption take place? Is it an OS service or does each application have to do it on its own?

Theft From Databases

- Databases have the most important data these days
- Even apart from the previous issues, how do you do encrypted search? Arithmetic searches?
- What about database joins?

Eavesdropping on Links

- Can an enemy listen to conversations?
- Which ones? What is the harm?

Authentication

- How do your employees authenticate?
- If you’re not using cryptographic authentication, e.g., FIDO2, why not. . . ?

Loss of Keys

- OK, we'll encrypt everything—but then what happens if a key is lost?
- How do you back up keys?
- How do you back up keys without more risk of them being compromised?

Crypto-Induced Unavailability

- Certificates expire—will resources become unavailable?
- If people override expiration or revocation, will an enemy exploit that?

Theft of Keys

- How you protect keys?
- Is that process too expensive?
- Is the enemy a cryptanalyst, a hacker, or a thief?
- How do you protect web site keys?

Legal Issues

- In some countries, use of encryption is restricted or regulated
- What does this do to your architecture?

Problem Summary

- We need encryption in many different places
- Communication keys are different than storage keys
- Some keys are more valuable than others if stolen
- Some keys are more valuable than others if the company loses them
- Some uses of encryption run into regulatory issues
- Let’s look at a few examples

The Risks of Stolen Keys

"At the trial of Jerry Whitworth, a spy who passed U.S. Navy keying information to the Russians, the judge asked the prosecution's expert witness: 'Why is it necessary to destroy yesterday's. . . [key] . . . list if it's never going to be used again?' The witness responded in shock: 'A used key, Your Honor, is the most critical key there is. If anyone can gain access to that, they can read your communications.'"

"The First Ten Years of Public-Key Cryptography"
Whitfield Diffie

Protecting Keys in Renaissance Venice

“Due to the instrumentality of ciphers for Venice’s intelligence pursuits, this department was ring-fenced with harsh regulations that included the death penalty when ciphers or their keys were mishandled.”

Venice’s Secret Service: Organizing Intelligence in the Renaissance
Ioanna Iordanou

Threats to Cryptosystems

“In the winter of 1940–1941, B.S.C. assigned CYNTHIA *the task of obtaining the Italian naval attaché in the embassy at Washington, Admiral Alberto Lais. Within a few weeks he was infatuated. When she was certain of her power over him, she told him directly that she wanted the naval code. Lais, despite his age and experience, agreed without any protest to betray his country for a woman.”*

The Codebreakers
David Kahn

Web Site Encryption

- Communication key—can be replaced at need
- Risk if lost: none; just reinstall
- Risk if unavailable: the web site is unavailable
- Risk if stolen: an enemy can spoof the web site—but they have to divert the traffic first, which is hard but by no means up to nation-state level

But. . .

- Some governments do want to spoof some websites, e.g., Iran impersonating gmail

☞ Yes, this is a serious threat model, but it has happened

- Understand your threats. . .

Web Site Encryption: Solution

- Store the key, unencrypted, on the server (or load-balancing front end)
- If either is compromised, you've got bigger problems!
- (Question: which is worse, load balancer compromise or web server compromise? Which is a more vulnerable target?)
- The key is generally not worth that much to an attacker (who can benefit more by phishing messages), but web site unavailability is a serious problem for the company

Key-Handling

- How is the private key handled?
- Small site: Generate it on that machine, send the public key to a CA, and request a certificate.
- Large site: Generate on a (secure!) server administration machine, get the public key and its certificate signed, securely ship out both keys to all web servers

☞ Large sites do central administration of all other aspects of their servers

Ship Around Private Keys?

- Conventional wisdom says not to do that
- Conventional wisdom is wrong. . .

☞ It’s best if all web servers in a cluster share the same key pair

☞ Getting a certificate signed requires a credential—why trust a web server with that?

☞ You’re doing all other administration for such machines remotely

☞ Some machines don’t have enough randomness to generate good private keys

Bad Randomness

“Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.”

John von Neumann (1951)

- We have good ways of generating *pseudorandom* numbers, but need a long-enough, unpredictable seed
- Sometimes, implementors get it wrong
- Other times, there just isn’t enough true-random data available

True-Random Seeds

- Hardware random number generator
- Reliance on quantum noise in the mic amplifier, the camera's sensor, air turbulence in disk drives, etc.
- But: do today's servers or smart cards have microphones, cameras, spinning disk drives, etc?

Quasi-True Random: Use Low-Order Bits

- Time of day in nanoseconds
- Interrupt timing
- Keystroke timing
- Mouse movements
- Network interpacket arrival time

☞ Not always available! Not always used properly!

Running Your Own CA

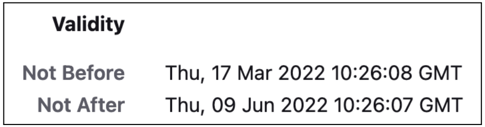
- Very large sites, e.g., Google may wish to run their own certificate authority
- Be your own root?
- No—the root has to be something widely trusted by browsers
- Instead: pay (a fair amount of money to) an existing root CA for a sub-root CA certificate
- How do you protect the signing key?

Protecting a CA Key

- Option 1: put it on the secure server administration machine
- Option 2: use an HSM
- Why an HSM?
- If that key is silently compromised, *all* of your web site certificates are compromised for a long time

Certificate Lifetimes

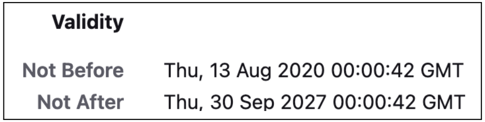
google.com



Validity

Not Before	Thu, 17 Mar 2022 10:26:08 GMT
Not After	Thu, 09 Jun 2022 10:26:07 GMT

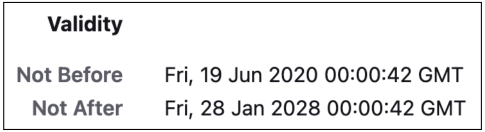
Google Intermediate CA



Validity

Not Before	Thu, 13 Aug 2020 00:00:42 GMT
Not After	Thu, 30 Sep 2027 00:00:42 GMT

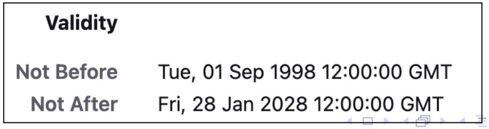
Google's Root CA



Validity

Not Before	Fri, 19 Jun 2020 00:00:42 GMT
Not After	Fri, 28 Jan 2028 00:00:42 GMT

CA (GlobalSign)



Validity

Not Before	Tue, 01 Sep 1998 12:00:00 GMT
Not After	Fri, 28 Jan 2028 12:00:00 GMT

Points to Note

- Different keys have different properties
- Different operational environments require different solutions

Employee Badge Keys

- Suppose employees have chip-enabled badges
- The chip must be programmed with a key pair and certificate
- The chip does cryptographic operations, including some that require random numbers
- These badges are used for multiple purposes: building entry, multi-factor authentication for login and VPN connections, digitally signing expense vouchers, decrypting encrypted email
- (Note: some large organizations do exactly this)
- How do we protect the certificate signing key?

Consequences of a Compromise

- Unauthorized physical access—serious
- Unauthorized login and VPN connections, if a password is also compromised—fairly serious (we use MFA *because* passwords are easily compromised
- Fake signatures—probably no worse than forged hardcopy signatures today

Points of Compromise

- Central corporate certificate-issuer
- Per-location certificate-issuing box?
- User badges

Protecting Signing Keys

- The root certificate is very precious—should be in an HSM
- Use it to create second-level certificates, one for each location
- Ideally, those should be in HSMs, too
- Use those to sign per-badge certificates

What About Small Sites?

- Some offices may be too small to have secure badging offices
- Solution: have a large site create the badge and ship it
- Is that secure?
- Probably—you're still trusting an employee to handle badging properly
- But how do you ship them securely?

Error Recovery

- What if an employee forgets their badge?
- What about lost badges?
- How they get in the building? How do they log in at their desks?
- (What if their badge is used to encrypt their home directory or their laptop's disk?)

Internal Network Encryption

- Should you encrypt internal network traffic?
- Yes—it helps protect you if your site has been penetrated
- Also: keeps visitors off your network: they can't connect to anything. . .
- But where do client keys come from? The user? The user's badge? The computer itself?
- N.B.: Must train employees not to click through encryption error messages, e.g., "Invalid certificate"
- And there are things you may not be able to encrypt easily, such as VoIP phones

Encrypting Data at Rest

- Encrypting some data, especially data that goes off-site—flash drives, laptops, off-site backups, etc.—is vital
- How do you ensure availability of the key?
- Unlike personal devices, enterprise data belongs to the enterprise

Encryption and Databases

- Simplistic approaches, e.g., encrypting each cell, don't work
- Modern encryption algorithms provide *semantic security*: no information leaks about the plaintext
- Two different encryptions of the same plaintext will produce different ciphertexts
- How do you database SELECTs? JOINs?
- There are encrypted databases, but they're not that simple

Encryption and Machine Learning

- Some data can't easily be kept encrypted because of the need to do machine learning on it
- In theory, one can do such things on encrypted data, using homomorphic encryption
- In practice, that's far too slow
- There are specialized approaches being tried, including by Facebook

We Need Back-up Keys

- But: we don’t need backup keys for communication sessions
- We don’t need backup keys for short-term copies of data, e.g., on a flash drive intended for data movement
- We don’t need backups for easily revocable/replaceable keys, e.g., some signing keys
- OS vendors offer backup password storage for disks—but *enterprise needs aren’t the same* as consumer needs
- How do you back up keys stored in HSMs?

Backup Strategies

- In many situations, the user-provided key is not the actual data encryption key (DEK)
- Instead, the DEK is randomly generated, and is encrypted by the provided key or password
- Permits easy change of password, and permits use of multiple keys, a primary and a backup
- Backup keys must be stored securely, reliably, and away from the protected data

☞ This last is especially important for off-site backups—and make sure you store two different copies of their keys, and in different spots

Backing Up HSMs

- HSMs should *never* export keys in the clear
- Solution: back them up to another HSM
- Yes, the two devices have to talk to each other first, so that the first knows to trust the second. . .

HSM Security

- HSMs are computers
- You speak to them via a specified protocol or API
- Is this API correct and secure? Most aren’t!
- How do you use an HSM with a cloud-resident VM web server? (Remember that by definition, clouds are someone else’s computer. . .)
- Do you have a networked HSM, for all of your web servers? How do you secure it?

HSM Security

- HSMs are computers
- You speak to them via a specified protocol or API
- Is this API correct and secure? Most aren't!
- How do you use an HSM with a cloud-resident VM web server? (Remember that by definition, clouds are someone else's computer. . .)
- Do you have a networked HSM, for all of your web servers? How do you secure it?

☞ At best, HSMs protect the key itself from theft. They do nothing to prevent abuse of the HSM itself.

Testing Backup Keys

- You never know if something works until you try it
- Do periodic tests of your keys
- (That’s even more true for backup media)

Managing and Tracking Keys

- Keys are important assets and need to be managed
- Certificates expire; that also needs management
- All of this needs to be tracked
- But: large sites already need to track other software issues

Key Escrow?

- “Prof. Bellovin, per the reading” I thought you opposed key escrow. . .

Key Escrow?

- “Prof. Bellovin, per the reading” I thought you opposed key escrow. . .
- I oppose *centralized, government-mandated and government-accessible key escrow*
- Cryptography is a *system*, and the problems often occur because of the systems nature
- Also: there is *never* a business need to escrow keys for communication sessions, and that’s a much harder problem

Differences

- You do not need a fancy protocol
- It is done with the cooperation of the subject
- A backup key could simply be a sealed envelope in a manager’s desk
- The trust relationships are completely different
- You’re dealing with a much smaller number of endpoints

Legal Issues

- In some jurisdictions, there are restrictions on use of encryption
- Example: there may be a requirement for providers to provide a back door for law enforcment, e.g., in Australia
- There are sometimes requirements for plaintext availability in certain industries, e.g., the financial sector in the US
- Brokerage houses are required to monitor communications with clients—but TLS 1.3 mandates *forward secrecy*

Operating at Scale

- Operating at scale is *different*
- You *must* use tools; you won't be able to cope otherwise
- You also won't get consistency
- Threats can be different, too

Questions?

(American kestrel, Columbia campus, January 30, 2023)