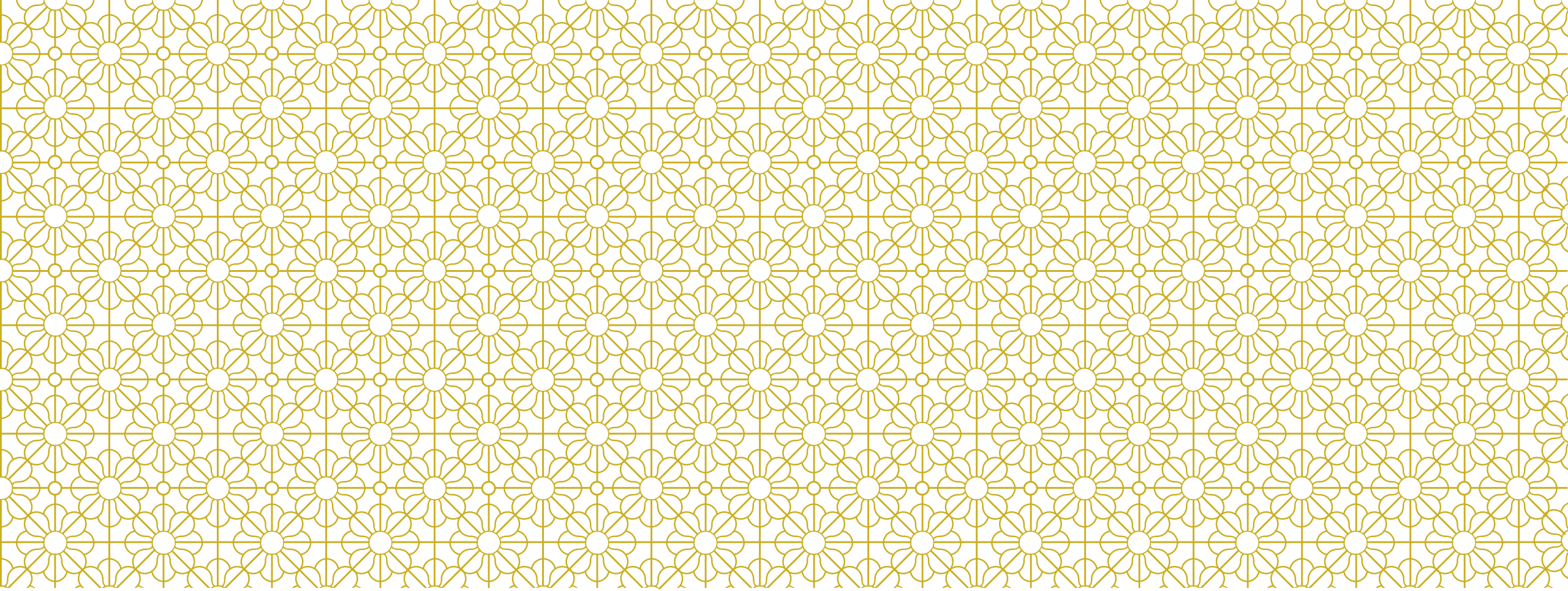

CONSIDERATIONS FOR USING SHORT-TERM CERTIFICATES

Nir, Fossati, Sheffer, Eckert

draft-nir-saag-star-01

WHAT IS THIS DRAFT ABOUT

This draft is about Short-Term Auto-Renewed certificates

While both of these properties describe STAR certificates, neither is the point

- The point is that there is no revocation information.
- Short-term is what we have make up for lacking the ability to revoke.
- Automatic reissue allows us to overcome the operational challenge of using short-term certificates.

The draft is intended to list the operational and security considerations for an environment with STAR certificates.

The intent is for this draft to become a BCP.

EXAMPLE #1: IPSEC VPN

Medium company VPN:

- 1 head office with a datacenter and two VPN gateways (for availability)
- 1 backup facility with its own two VPN gateways.
- 4 R&D centers with several networks and a VPN gateway each.
- 3 regional sales offices, each with its own datacenter and gateways.
- 30-60 sales offices in different countries and/or US states or regions, each with 5-30 people and a VPN gateway
- 100-1000 home offices with smaller VPN gateways
- 500-10,000 VPN clients installed on company or BYOD laptops or phones.

That's many hundreds of gateways and thousands of clients.

- While the clients mostly connect to datacenter gateways, any of the VPN gateways may connect to any of the others based on topology.

EXAMPLE #1: SECURITY CHALLENGES

Stop intruders from connecting to the VPN.

Prevent a rogue gateway or client from impersonating another.

Allow IPsec traffic from any gateway only if the source addresses belong to the network it is protecting.

All of these require authentication between hosts.

EXAMPLE #2: SOFTWARE-DEFINED STORAGE

Dozens to hundreds for Data Servers, each with several storage devices holding anywhere from 5-100 TB each.

Data is mirrored ("RAID-1") or uses parity ("RAID-5").

Anywhere from 1 to hundreds of "Data Clients" (application servers) that have a driver installed. These may or may not be co-located with the Data Servers.

Virtual volumes are "mounted" on the Data Clients. Read and write access uses networking, optionally with encryption, either IPsec or TLS.

A controller (replicated, of course) runs the whole thing:

- It allocates the volume space on the Data Servers, and tell them where to store the second copy.
- Mounting a volume on a Data Client means that the controller sends it a mapping of the volume
- It manages balancing and recovery from failures.

EXAMPLE #2: SECURITY CHALLENGES

There are all sorts of things we might want to accomplish in a system like this:

- Hosts that are not defined as Data Clients should not be able to access or modify data.
- Data Clients should not be able to access or modify volumes that are not mounted to them.
- Data Clients with read-only access to a volume should not be able to modify data.
- An attacker should not be able to impersonate a controller or a data server, allowing them to read, modify, or fake the data.

None of this can be accomplished without authentication between the services.

AUTHENTICATION

While it is possible to authenticate communications using pair-wise shared secrets, you always end up reverting to certificates.

There are really three sources for you to get certificates for your hosts or services:

- There is the web PKI with multiple CAs selling or giving away certificates.
- There is the corporate CA (usually part of a domain)
- You can roll your own for your product.

You almost always roll your own. Why?

- The Web PKI is geared for web servers and sometimes for email users. They don't work so well for a bunch of internal servers that don't communicate over the Internet.
- While the corporate CA can do anything, there is often little expertise or willingness to issue hundreds of certificates for some vendor system.
- If you roll your own CA, your controller can issue the certificates that you need when you need them with little intervention.

REVOCATION

When rolling our own PKI, we'd rather not have to deal with revocation:

- Revocation adds complexity.
  - This adds a bunch of failure modes.
- Revocation takes time, due to caching and process time.
- Revocation slows down connection establishment.
- Issuing blob-1 and then blob-2, which tells the RP that blob-1 is still valid does not make sense, especially when issuing blob-1 and blob-2 require similar effort.
  - With automated certificate issuance, issuing a new certificate is as easy as issuing a CRL or OCSP status.

REVOCATION ALTERNATIVE

Stopping a rogue node with a certificate is still necessary.

Our solution is to stop renewal.

To get equivalent security properties, we propose to make the lifetime of the certificates short.

- As short as the typical time it takes us to issue new revocation information plus the time it takes for RPs to see it and use it.
- Days or hours rather than months.

To deal with the administrative nightmare of issuing hundreds of certificates every day, automatic renewal must be used.

THE DRAFT

Admittedly, the draft requires work.

- Use case examples can be added rather than just hinted at.
- Many suggestions we've heard in previous meetings should be discussed and if there's consensus, added.
- More discussion is needed about the requirements to allow us to consider a system with STAR certificates and no revocation information as secure enough.
- We probably need more about pitfalls in using short-term certificates.

We hope to do this work in this working group

ANTICIPATED QUESTIONS

Is this for the Web?

- No. At least, not necessarily.

Maybe we need an extension for “no revocation information”?

- I don’t think so.
- But if the group wants one, I think it should be a separate document.
- This should remain a BCP.

Doesn’t TLS 1.3 with client certificate OCSP stapling make this unnecessary?

- I don’t think so. Client-side OCSP stapling does not solve all of the problems:
  - It doesn’t solve the complexity.
  - It doesn’t solve the requirement for an always-on revocation server.
  - It reduces but does not eliminate the additional failure modes.