Real Time Credential Validation

Secure, Efficient Permissions Management

Contact us at:
CoreStreet Ltd.
One Alewife Center
Suite 200
Cambridge, MA 02140

info@corestreet.com
Tel: 617-661-3554

Credentials

Credentials are used in many different contexts to provide identification and privileges for their bearers. For example, Alice may have a credential in the form of a student identification card at her university to permit entry into the library and gym. Credentials are also used in secure digital environments. As another example, Bob may have a digital certificate on an electronic "smart card" that allows him to access his company's computer network and storage lockers.

Authentication

The primary function of both types of credentials is *authentication:* proving that Alice and Bob are who they say that they are. To perform authentication, Bob presents his credential along with some supplemental identifying information. This identification is typically based on a secret known only to the bearer ("something you know") or some physical characteristics ("something you are"), or even both.

A librarian can authenticate Alice by checking that she has a valid student ID with a picture that matches her face. Bob provides his credential and a secret PIN to authenticate himself to the company storage lockers.

If the credentials are reasonably hard to forge and the supplemental identifying information is correct, Alice and Bob can be authenticated with a high degree of certainty. Someone like the librarian (referred to as the *relying party*) can authenticate Alice's identity based on her two-year-old student ID, and can do it without calling the registrar or checking any databases. Authentication has some limitations, however. Authentication establishes that this particular woman is Alice, and that she was a student two years ago, but it does not prove that she is still a student with library privileges.

Validation

To establish someone's *current* status and privileges, a relying party needs to perform a second function called *validation.* While authentication answers the question *"Are you who you say you are?"* validation answers the more sensitive question, *"Are you currently allowed to do what you are trying to do?"* The storage locker needs to know that Bob is really Bob (authentication), and that Bob is currently authorized to open the locker (validation). Authentication and validation are both required for secure access control.

While it is straightforward to perform authentication at the point of contact without accessing any centralized databases or networks, validation is typically a more difficult problem. Classical validation is performed with one of two general techniques.

Revocation Lists

The first classic validation solution is to create a master reference list of all credentials that have been issued, with information about which ones are currently canceled (or *revoked*). This master list must be published frequently (daily, weekly) along with proof of authenticity (seal, digital signature, etc.) and a copy must be given to every relying party. For example, the librarian might have a printed, notarized list of every obsolete student ID number, and Alice may enter only if her number is not on the list.

List-based validation can be performed extremely quickly by a relying party without needing to talk to a separate authority with every transaction (*offline validation*). Unfortunately, these master lists can become extremely large and unwieldy for many applications, and it is frequently not practical to transfer a new list to every potential relying party every day. For example, Bob's company may have dozens of storage lockers, with no network connections. It would be impractical for someone to walk around daily to update each locker with a new list of revoked digital certificates (a *Certificate Revocation List,* or *CRL*).

CORESTREET
REAL TIME CREDENTIALS

Real Time Credential Validation

Secure, Efficient Permissions Management

Online Validation

The second classic validation solution requires that a relying party ask a central, trusted authority whether each set of credentials is still authorized. Alice's librarian may do this by checking the central university database to see whether Alice has library privileges, or could even call the registrar's office to check their records.

This *online* scheme allows immediate validation without requiring the distribution of massive daily lists to every relying party. Unfortunately, this scheme has problems with *availability, security* and *scalability.*

First, the online scheme requires that the relying party and the central authority have a dependable connection to talk to each other. Any problems with communication will cause a disruption that prevents validation.

This scheme also introduces the security risk that a counterfeit "authority" may intercept the validation request from the relying party and provide a false approval response. Alice's friend Charlie may pretend to be the registrar when the librarian picks up the phone, thereby allowing Alice invalid library access. This risk can be reduced by various techniques such as providing trusted authorities with their own secret credential information, but this still runs the risk of compromise. For example, if Charlie can trick one of the registrars into divulging his secret credential information, Charlie will still be able to fool the librarian into accepting a false validation.

Finally, the online scheme has inherent limits of scale. If there are only a small number of relying parties in close geographic proximity, then a central authority will be able to talk to all of them and provide prompt validation responses. If there are a large number of relying parties and a large population of credentials to validate, the number of trusted authorities will need to grow significantly. If Alice travels to the overseas campus in Europe, the library there will need to validate her credential even if the connections to America have problems with cost, speed, time zones, etc.

Real Time Credentials

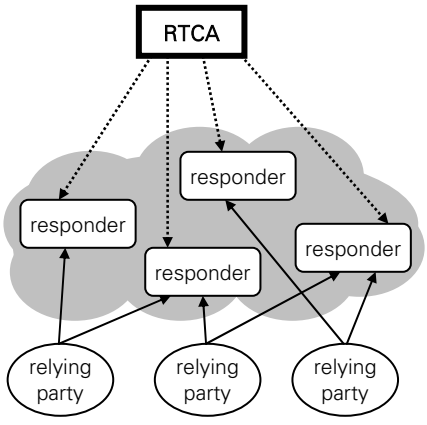
CoreStreet has developed an alternate validation solution called *Real Time Credentials* (*RTC*) that eliminates the problems of the two classic credential validation schemes while maintaining the key advantages of each:

- Offline validation using published, verifiable proofs of credential status
- Short messages, fast communication

An RTC solution is based around the concept of a *Credential Validation Proof*, which is a verifiable assertion about a credential's current status. The important characteristics of a single RTC proof are:

- **Pre-generated** – created and published periodically, not based on specific requests
- **Individual** – each proof corresponds to a single person's credential
- **Small** – from as small as 16 bytes to a few hundred bytes per proof
- **Verifiable** – can tell a forged or tampered proof from a real one
- **Bounded** – usable for a specific window of time

CoreStreet has patented several different algorithms to represent credential validation proofs based on these characteristics, but the high-level scheme is shown in the following diagram.



Real Time Credential Validation

Secure, Efficient Permissions Management

A single, centralized *Real Time Credential Authority* (*RTCA*) manages validation information for a set of credentials. The RTCA periodically creates and publishes RTC proofs for each of these credentials. Each proof will say whether one credential is still valid, and optionally provide more detailed information about the current attributes or privileges of the credential bearer.

Real Time Credentials allow a relying party to validate a user through the following sequence of steps:

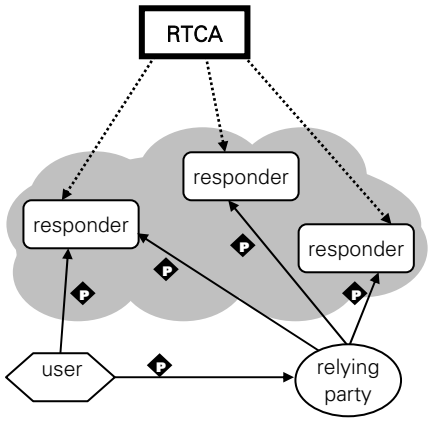
1. Receive the user's credential
2. Verify the integrity of the credential (digital signatures, watermarks, holograms, etc.)
3. Check supplemental authentication factors (PIN number, photograph, public key challenge, biometrics, etc.)
4. Receive the RTC validation proof for the user's credential
5. Verify the integrity of the proof
6. Verify that the proof is current

The first three steps perform authentication of the user, and the last three steps perform validation of the credential using an RTC proof. As long as the relying party can get the current proof, it can perform all of these operations without making any real time connections to any central, trusted authorities.

RTC proofs are public, verifiable and tamper-evident, so they can be published widely. They can even be safely re-published by a second entity called a *responder*. Since the responders are only relaying proofs that were pre-generated by the RTCA, they don't need to be individually trusted or secured. This means that a relying party can accept a proof coming from any responder, regardless of affiliation.

In addition, the relying party can even accept a validation proof that comes from the user that is being validated. This means that a user can store his own RTC proof each morning and then hand that proof to a relying party, who can perform immediate authentication and validation without making any online connections.

The following diagram shows different ways that a relying party could receive the current validation proof (P) for a user's credential.



Note that in all of these cases, the relying party is receiving a small, individualized proof, and it processes the proof without needing to trust the carrier that delivers it. Even a relying party without any network connections can use this solution by receiving a proof from the user himself.

Example

Bob has a secure smart ID card that contains his X.509 digital certificate. He arrives at work in the morning and uses his ID card at the front entrance. The smart door lock performs standard authentication to establish his identity, then uses its network connection to request an RTC validation proof from a nearby responder.

That responder has a directory of all employee validation proofs published by the company's RTCA for the day. The responder looks up Bob's proof in a couple of milliseconds and returns it to the door.

The door controller uses this validation proof to prove that Bob is still an employee and that he has permission to enter that building. The door also places a copy of Bob's validation proof on his smart card.

Later, when Bob needs to retrieve some tools from the non-networked secure storage locker, he inserts his card into the

lock. The locker authenticates Bob based on his credential and then retrieves the stored validation proof from the card. The locker examines the validation proof along with the credential to prove that Bob is currently permitted to access that locker before unlocking the door.

At no time does any machine need to make a request to the secure RTCA or any other trusted entity.

Comparison

The Real Time Credential validation solution offers dramatically superior availability, security, and scalability compared to the two classic validation approaches.

Since RTC proofs are tamper-evident, they can be published to a wide network of inexpensive responders. This distributed architecture greatly increases the service availability since every relying party should be able to find some responder that can provide it with a prompt response.

This distributed network of RTC responders does not have any control functionality and does not hold any secrets, so the responder network can easily grow without compromising security. Responders don't require hardware security modules, vaults, or guards. RTC solutions can easily be configured so that no external communications are ever allowed to the sensitive RTCA, which prevents any network attacks that could compromise sensitive information.

Finally, the RTC solution allows unlimited scalability, since cheap responders can be easily added to increase the number of credentials and relying parties that can be handled. Responders are only mirroring pre-generated information created by the RTCA, so they perform very little work and are only constrained by their communications bandwidth.

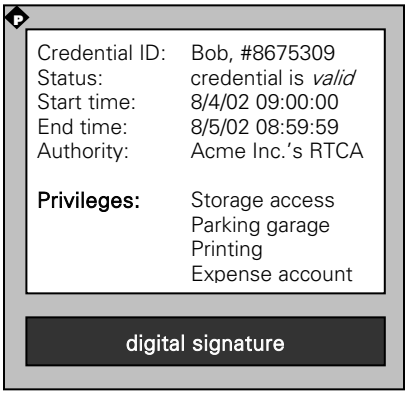
RTC Implementations

The previous sections provide the high-level description of the Real Time Credential technology. CoreStreet's RTC Validation Authority delivers a complete credential validation solution that includes the RTCA, RTC responders, and relying party libraries. The RTC Validation Authority implements validation through proofs based on two different security formats. These formats offer equivalent functional and security characteristics, but differ in their operational advantages.

Digital Signatures

The first RTC proof format is structured as a set of status information that is digitally signed using the RTCA's private key. The body of the proof includes information about its starting and ending time as well as the status of the credentials and any associated privileges or attributes. The digital signature is based on standard digital signature algorithms such as RSA, DSA or ECC.

The following diagram shows the contents of a hypothetical digitally signed RTC proof.



The RTC Validation Authority publishes this type of proof to provide validation for any type of credential. The exact format of this proof is specified in the Online Certificate Status Protocol (OCSP) standard published by the IETF as RFC 2560. A proof of this format is typically between 150 and 350 bytes in size, and is compatible with all existing certificate credentials.

A relying party can trust this type of proof based on the integrity of the digital signature and the corresponding RTCA private key.

This RTC proof format is fully compatible with any relying parties that are using the

OCSP standard for performing validation, but it maintains the distinct RTC advantages of availability, scalability, and security which traditional online validation OCSP implementations lack. This allows the OCSP protocol to be used to manage millions of certificate credentials over a global physical environment.

Validation Tokens (VTokens)

The second RTC proof format implemented by the RTC Validation Authority is based around secure one-way hashing functions such as the SHA-1 or MD5 standards. VTokens offer even greater performance and scalability for environments with high-end requirements.

To use VToken technology, each user's credential is embedded at creation time with a unique VToken *anchor* value (formally referred to as V_Ω). For each of these anchor values, the RTCA remembers a corresponding secret VToken *seed* value (formally referred to as V_0).

When a credential is VToken-enabled in this manner, the RTCA issues periodic validation proofs that are as small as 16 bytes by using its secret V_0 seed to generate a special VToken *proof* (V_i). A relying party can inspect this small proof in a fraction of a millisecond to determine whether the proof and its corresponding credential are currently valid.

If a user's credential is revoked, then the RTCA can issue a special revocation proof that can be used in the same manner as the validation proof.

The VToken format allows for proofs that are a tenth of the size of digital signature proofs. These proofs can also be generated and evaluated in a hundredth of the time of digital signatures.

Format Selection

RTC proofs using either digital signature or VToken formats can offer secure credential validation for a variety of applications.

Digital signature proofs offer the simplest integration based on their compatibility with existing protocols and standards.

Depending on signature algorithm, key length, and privilege management, digital signature proofs may be between 150 and 350 bytes, and can typically be processed and verified in under ten milliseconds on a typical computer. A digital signature solution offers excellent scalability to about ten million independent credentials and hundreds of responders.

VToken proofs provide even greater scalability and performance for environments without requirements for legacy compatibility. Depending on one-way hashing algorithm and protocol selection, VToken proofs may be between 16 and 100 bytes in size and can be generated and evaluated in less than one millisecond. A VToken solution can easily support hundreds of millions to billions of credentials along with tens of thousands of responders.

Using either format, a Real Time Credential solution provides for unequaled availability, scalability, and security for mission-critical credential validation.