# Credentials Management for High-Value Transactions

Glenn Benson[1], Shiu-Kai Chin[2], Sean Croston[1], Karthick Jayaraman[2], and Susan Older[2]

[1] JP Morgan Chase {*glenn.benson,sean.b.croston*}*@jpmchase.com*
[2] EECS Department, Syracuse University, Syracuse, New York 13244
{*skchin,kjayaram,sbolder*}*@syr.edu*

**Abstract.** Partner key management (PKM) is an interoperable credential management protocol for online commercial transactions of high value. PKM reinterprets traditional public key infrastructure (PKI) for use in high-value commercial transactions, which require additional controls on the use of credentials for authentication and authorization. The need for additional controls is met by the use of partner key practice statements (PKPS), which are machine-readable policy statements precisely specifying a bank's policy for accepting and processing payment requests. As assurance is crucial for high-value transactions, we use an access-control logic to: (1) describe the protocol, (2) assure the logical consistency of the operations, and (3) to make the trust assumptions explicit.

**Keywords:** authentication, authorization, protocols, trust, logic

## 1 Introduction

Authorizing online high-value commercial transactions requires a higher level of diligence when compared to consumer or retail transactions. A single high-value transaction may involve the transfer of hundreds of millions of dollars. The inherent risk associated with wholesale online banking compels many banks to require additional security beyond authenticating users at login time. Additional security often takes the form of tighter controls and limits on the use of credentials. Ultimately, each bank trusts itself more than any other entity. This naturally leads to the practice of banks issuing their own credentials. Historically, a cash manager of a corporation would hold separate credentials from each bank with which he or she deals. While this serves the needs of commercial banks, as corporations want to simultaneously hold accounts in multiple banks, the insistence upon and proliferation of unique credentials is viewed by customers as poor service. Hence, it is increasingly important for global financial services providers, such as JP Morgan Chase, to offer credentials that: (1) are interoperable to provide customer convenience, and (2) meet the needs of high-value commercial transactions in terms of authentication, authorization, and liability.

Traditional Public Key Infrastructure (PKI) credentials while interoperable, alone are insufficient to surmount the following obstacles inherent to the use of interoperable credentials in high-value transactions:

1. *Autonomy*: Interoperability and autonomy are in tension with each other. An implication of interoperability is the need to allow audits. For example, say Second Bank is contemplating recognizing credentials issued by First Bank. Second Bank would understandably want to audit First Bank's practices as a credential issuer against Second Bank's policies. Understandably, First Bank would be reluctant to agree to audits of its operations by competitors such as Second Bank.
2. *Liability*: Non-bank issuers of PKI credentials neither want, nor are in a position to accept, liability for failed high-value transactions. One way around this is for a bank to issue its own credentials to limit risk and to recognize only the credentials it issues; however, the solution is not interoperable by definition.
3. *Expense*: If commercial banks were to recognize non-bank certificate issuers for high-value commercial transactions, then commercial banks would need to be connected to the non-bank certificate issuers. This is an added operational expense for banks, which is another barrier to achieving interoperability.

In this paper, we describe an interoperable certificate management protocol called *partner key management* (PKM). PKM is designed to address the three obstacles to interoperability of credentials in high-value transactions described above. Under the PKM model, each bank publishes a *partner key practice statement* (PKPS), which is a machine readable document that describes the bank's policy for accepting interoperable credentials. PKM enables each bank to avoid liability on transactions executed at any other bank, while preserving credential interoperability. Furthermore, PKM supports a general validation model, where each corporation need only connect to the credential issuers to which it subscribes. Moreover, we describe the certificate management protocol using an access-control logic to prove its logical consistency and also to make the underlying trust assumptions explicit.

The rest of this paper is organized as follows. Section 2 presents the PKM model, PKPS, and sender validation. Section 3 defines the syntax, semantics, and inference rules of the logic used to describe and reason about PKM. Section 4 is an overview of how key parts of PKPS are expressed in the logic. Section 5 provides an extended example describing and analyzing the operation of PKM. Related work is briefly discussed in Section 6. We offer conclusions in Section 7.

## 2 Partner Key Management

### 2.1 Credentials Registration

The PKM model focuses on authorization to use a credential as opposed to secure distribution of a credential. As an analogy, consider mobile phone distribution logistics. A user may purchase a mobile phone from any distributor. At the time that the user physically acquires the phone, the telecom operator does not know the user's identity and does not allow use of the phone. Subsequently, the user and the telecom operator agree to terms of use; and the mobile phone operator authorizes the phone's connection to the telecom network. In the PKM model,

the credential plays the role of the phone, and the bank plays a similar role to the telecom operator.

In PKM, the user first obtains a credential from a credential distributor. The credential distributor has the responsibility to distribute 'secure' credentials under a definition of security defined by the operator. For example, one operator may only distribute certificates on secured USB devices, while another operator may distribute software for self-signed certificates. After obtaining a credential, the user submits a request to each of his or her banks to allow use of the credential. On this step, the bank has two responsibilities. First, the bank must securely assure itself of the user's true identity. Second, the bank must examine the credential to determine if the credential meets the bank's standards. For example, some banks may prohibit credentials other than certificates that reside in a secured hardware token. If the bank accepts the credential, then the bank authorizes the credential to represent the user. The user may use the same credential with multiple banks by appropriately registering the credential with the respective banks. The authorization process may vary between the banks. Each bank may have its own operational policy governing the conditions in which it accepts the credentials based upon the bank's published operating rules.

In effect, the credentials are interoperable, and banks have the liberty to follow their own procedure for accepting the credentials and allowing users to employ those credentials. The result is an infrastructure that allows the possibility of interoperability without mandating interoperability. If two banks agree to accept a single credential, then that credential would interoperate between the two banks. No bank needs to rely upon any other bank or external credential provider.

### 2.2 Partner Key Practice Statement

Banks participating in the PKM model publish an XML document called the *Partner Key Practice Statement* (PKPS), which is written using WS-Policy [1]. A PKPS defines how a corporation and a bank agree to work together, as governed by their mutually agreed upon security procedures. The corporation and the bank have the freedom to impose almost any conditions to which they mutually agree, provided that the conditions do not require unsupportable programming logic. The list below presents some examples types of information that may appear in a PKPS:

1. *Credential Media:* The definition of the credential media may mandate a smart card, USB token, HSM, FIPS-140-2, or a software credential.
2. *Credential Provider:* This item contains the list of credential providers to which the corporation and the bank mutually subscribe. Example providers are third party trusted providers, self-signed certificates, the corporation's, or the bank's own infrastructure.
3. *Revocation:* The revocation definition describes the type of permissible credential revocation mechanism, e.g., certificate revocation list (CRL), online certificate status protocol (OCSP) [2], etc. The revocation definition also describes the party responsible for enforcing credential revocation; and it describes any specific usage practice. For example, the revocation mechanism may mandate that the recipient of a signature validate a CRL signed by a particular party.

4. *Timestamp:* The timestamp definition defines timestamp rules and the timestamp provider, if any. The timestamp definition may specify a real-time threshold value. The recipient must ensure that it receives and validates a signature before the threshold timelimit after the timestamp. For example, a six hour threshold value means that the recipient must validate a signature before six hours expires after the timestamp.
5. *Signature Policy:* The PKPS can specify the number of signatures required for a specific type of transaction, and the roles of signatories. An example of a signature policy is one which requires both an individual signature and a corporate "system" signature in order to consider either signature as valid.
6. *Credential Technology:* A certificate that supports the X.509 standard is an obvious choice for interoperability. However, additional technologies such as the portable security transaction protocol (PSTP) [3] exist, and the PKPS may specify alternative technologies.

The security requirements mutually agreed to by the bank and the corporation are reflected in a specific PKPS, or possibly a list of PKPSs. The security requirements may mandate that the corporation must attach the PKPS on each signed transaction in order to consider any signature valid.

### 2.3 Revocation

This paper presents three example validation models. A bank's PKSP should define the model that a particular bank allows.

1. **Receiver validation:** The receiver validation model is typically used in a PKI model. First, Alice submits a signed transaction to the bank. Upon receipt, the bank validates Alice's signature against a CRL or OCSP responder managed by the certificate provider.
2. **Sender validation without evidence:** Alice submits signed transactions to the bank, but the bank performs no revocation check. Alice's company and the bank manage Alice's credential using mechanism outside the scope of the signed transaction.
3. **Sender validation with evidence:** Alice submits her certificate to an OCSP responder, and obtains a response signed by the OCSP responder. Alice signs the transaction and the OCSP response, and then submits to the bank. The bank validates both Alice's signature and the OCSP responder's signature. If the bank finds no error, then the bank accepts the transaction.

Each bank has the opportunity to allow any of the three example models, or build its own variant model. Multiple banks may all accept the same credential from Alice, while requiring different revocation models.The second model, sender validation without evidence, merits further discussion. If Alice proves to be an untrustworthy person, then Alice's company reserves the right to disable Alice's credential. For example, if Alice has a gambling problem, then authorized representatives of Alice's company should contact each of its banks with the instruction to stop allowing Alice's credential. Another use case which also results in credential disabling, is one where Alice contacts each bank because she suspects that her own credential was lost or stolen.

An OCSP responder, or a certificate revocation list is merely a revocation mechanism optimized for scalability. As opposed to requiring the Alice's company to contact each of its banks, an OCSP responder or Certificate Revocation List provides a centralized repository which handles certificate revocation. The advantage of the OCSP responder or certificate revocation list is scalability as opposed to security. If Alice were authorized to transact on accounts at hundreds or thousands of banks, then the second model (sender validation without evidence) would not be practical. However, in practice, wholesale banking does not need such enormous scalability. Rather, Alice typically works with just a handful of banks. Although Alice's company may find the credential disabling process to be relatively tedious because the company needs to contact each of the banks in the handful, we normally find that corporations employ the credential disabling process relatively infrequently.

In practice, corporations tend to contact each of their banks whenever a user's credential changes status, even if the bank happens to use the traditional receiver validation model. In fact, some banks require immediate notification of such events in their operating model. Intuitively, if the corporation ceases to trust Alice to authorize high-value transactions, then the corporation probably wants to contact each of its banks directly.

Both the second and the third models assume sender validation, as opposed to receiver validation. An advantage of sender validation is that it better handles expense. Suppose, for example, a corporation agrees to the services of a new credential distributor. Credential interoperability encourages a dynamic market by allowing the corporation the freedom to choose any acceptable credential distributor. In the receiver validation model, the corporation could not use that credential with its bank until the bank agrees to build an online connection to the credential distributor's OCSP responder or certificate revocation list. In the sender validation models, on the other hand, the corporation may immediately use the credential with the bank without waiting for the costly and possibly slow technology development process.

## 3 An Access-Control Logic and Calculus

We use an access-control logic to describe and reason about the validity of acting on payment instructions. This section introduces the syntax, semantics, and inference rules of the logic we use.

### 3.1 Syntax

*Principal Expressions* Let $P$ and $Q$ range over a collection of principal expressions. Let $A$ range over a countable set of simple principal names. The abstract syntax of principal expressions is:

$$P ::= A \ / \ P\&Q \ / \ P \mid Q$$

The principal $P\&Q$ ("$P$ in conjunction with $Q$") is an abstract principal making exactly those statements made by both $P$ and $Q$; $P \mid Q$ ("$P$ quoting $Q$") is an abstract principal corresponding to principal $P$ quoting principal $Q$.

*Access Control Statements* The abstract syntax of statements (ranged over by $\varphi$) is defined as follows, where $P$ and $Q$ range over principal expressions and $p$ ranges over a countable set of *propositional variables*:

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \equiv \varphi_2 \mid P \Rightarrow Q \mid P \text{ says } \varphi \mid P \text{ controls } \varphi \mid P \text{ reps } Q \text{ on } \varphi$$

Informally, a formula $P \Rightarrow Q$ (pronounced "$P$ speaks for $Q$") indicates that *every* statement made by $P$ can also be viewed as a statement from $Q$. A formula $P$ controls $\varphi$ is syntactic sugar for the implication $(P \text{ says } \varphi) \supset \varphi$: in effect, $P$ is a trusted authority with respect to the statement $\varphi$. $P$ reps $Q$ on $\varphi$ denotes that $P$ is $Q$'s delegate on $\varphi$; it is syntactic sugar for $(P \text{ says } (Q \text{ says } \varphi)) \supset Q \text{ says } \varphi$. Notice that the definition of $P$ reps $Q$ on $\varphi$ is a special case of controls and in effect asserts that $P$ is a trusted authority with respect to $Q$ saying $\varphi$.

### 3.2 Semantics

Kripke structures define the semantics of formulas.

**Definition 1.** *A* Kripke structure $\mathcal{M}$ *is a three-tuple* $\langle W, I, J\rangle$*, where:*

- $W$ *is a nonempty set, whose elements are called* worlds*.*
- $I : \boldsymbol{PropVar} \to \mathcal{P}(W)$ *is an* interpretation *function that maps each propositional variable* $p$ *to a set of worlds.*
- $J : \boldsymbol{PName} \to \mathcal{P}(W \times W)$ *is a function that maps each principal name* $A$ *to a relation on worlds (i.e., a subset of* $W \times W$*).*

We extend $J$ to work over arbitrary *principal expressions* using set union and relational composition as follows:

$$J(P\&Q) = J(P) \cup J(Q)$$
$$J(P \mid Q) = J(P) \circ J(Q),$$

where

$$J(P) \circ J(Q) = \{(w_1, w_2) \mid \exists w'.(w_1, w') \in J(P) \text{ and } (w', w_2) \in J(Q)\}$$

**Definition 2.** *Each Kripke structure* $\mathcal{M} = \langle W, I, J\rangle$ *gives rise to a function*

$$\mathcal{E}_{\mathcal{M}}[\![-]\!] : \boldsymbol{Form} \to \mathcal{P}(W),$$

*where* $\mathcal{E}_{\mathcal{M}}[\![\varphi]\!]$ *is the set of worlds in which* $\varphi$ *is considered true.* $\mathcal{E}_{\mathcal{M}}[\![\varphi]\!]$ *is defined inductively on the structure of* $\varphi$*, as shown in Figure 1.*

*Note that, in the definition of* $\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!]$*,* $J(P)(w)$ *is simply the image of world* $w$ *under the relation* $J(P)$*.*

### 3.3 Inference Rules

In practice, relying on the Kripke semantics alone to reason about policies and behavior is inconvenient. Instead, inference rules are used to manipulate formulas in the logic. All logical rules must be sound to maintain consistency.

**Definition 3.** *A rule of form* $\dfrac{H_1 \cdots H_n}{C}$ *is sound if for all Kripke structures* $\mathcal{M} = \langle W, I, J\rangle$*, if* $\mathcal{E}_{\mathcal{M}}[\![H_i]\!] = W$ *for each* $i \in \{1, \ldots, n\}$*, then* $\mathcal{E}_{\mathcal{M}}[\![C]\!] = W$*.*

The rules in Figures 2 and 3 are all sound. If sound rules are used throughout, then the conclusions derived using the inference rules are sound, too.

$$
\begin{aligned}
\mathcal{E}_{\mathcal{M}}[\![p]\!] &= I(p) \\
\mathcal{E}_{\mathcal{M}}[\![\neg\varphi]\!] &= W - \mathcal{E}_{\mathcal{M}}[\![\varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \wedge \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \vee \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!] \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] &= (W - \mathcal{E}_{\mathcal{M}}[\![\varphi_1]\!]) \cup \mathcal{E}_{\mathcal{M}}[\![\varphi_2]\!] \\
\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \equiv \varphi_2]\!] &= \mathcal{E}_{\mathcal{M}}[\![\varphi_1 \supset \varphi_2]\!] \cap \mathcal{E}_{\mathcal{M}}[\![\varphi_2 \supset \varphi_1]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \Rightarrow Q]\!] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ says } \varphi]\!] &= \{w | J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\![\varphi]\!]\} \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ controls } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![(P \text{ says } \varphi) \supset \varphi]\!] \\
\mathcal{E}_{\mathcal{M}}[\![P \text{ reps } Q \text{ on } \varphi]\!] &= \mathcal{E}_{\mathcal{M}}[\![P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]\!]
\end{aligned}
$$

Fig. 1: Semantics

## 4 Expressing Statements and the PKPS in Logic

With the definition of the syntax and semantics of access-control logic, we provide an introduction to expressing actual payment instructions and the PKPS in logic.

*Statements and Certificates:* Statements and requests are made by principals. Requests are logical statements. For example, say Alice wants to transfer $\$10^6$ dollars from $acct_1$ to $acct_2$. If $\langle transfer\ 10^6, acct_1, acct_2\rangle$ denotes the proposition *it is justifiable to transfer* $\$10^6$ *from* $acct_1$ *to* $acct_2$, then we can represent Alice's request as $Alice$ says $\langle transfer\ 10^6, acct_1, acct_2\rangle$. Credentials or certificates are statements, usually signed with a cryptographic key. For example, assume we believe $K_{CA}$ is the key used by certificate authority $CA$. With this belief, we would interpret a statement made by $K_{CA}$ to come from $CA$. In particular, if $K_{CA}$ says $(K_{Alice} \Rightarrow Alice)$, we would interpret this public key certificate signed by $K_{CA}$ as having come from $CA$.

*Authority and Jurisdiction:* Jurisdiction statements identify who or what has authority, specific privileges, powers, or rights. In the logic, jurisdiction statements usually are controls statements. For example, if Alice has the right to transfer a $\$10^6$ dollars from $acct_1$ to $acct_2$, we say $Alice$ controls $\langle transfer\ 10^6, acct_1, acct_2\rangle$. If Alice has jurisdiction on $\langle transfer\ 10^6, acct_1, acct_2\rangle$ and Alice requests $\langle transfer\ 10^6, acct_1, acct_2\rangle$, then the *Controls* inference rule in Figure 3 allows us to infer the soundness of $\langle transfer\ 10^6, acct_1, acct_2\rangle$.

$$
\frac{Alice \text{ controls } \langle transfer\ 10^6, acct_1, acct_2\rangle \qquad Alice \text{ says } \langle transfer\ 10^6, acct_1, acct_2\rangle}{\langle transfer\ 10^6, acct_1, acct_2\rangle.}
$$

*Proxies and delegates* Often, something or somebody makes the requests to the guards protecting the resource on behalf of the actual principals, who are the sources of the requests. In an electronic transaction, a cryptographic key is used as a proxy for a principal. Recall that $K_{CA}$ says $(K_{Alice} \Rightarrow Alice)$ is a public key certificate signed with the public key $K_{CA}$ of the certification authority.

$$\textit{Taut}\ \frac{}{\varphi}\quad \text{if } \varphi \text{ is an instance of a prop-logic tautology}$$

$$\textit{Modus Ponens}\ \frac{\varphi \quad \varphi \supset \varphi'}{\varphi'} \qquad \textit{Says}\ \frac{\varphi}{P\ \mathsf{says}\ \varphi}$$

$$\textit{MP Says}\ \frac{}{(P\ \mathsf{says}\ (\varphi \supset \varphi')) \supset (P\ \mathsf{says}\ \varphi \supset P\ \mathsf{says}\ \varphi')}$$

$$\textit{Speaks For}\ \frac{}{P \Rightarrow Q \supset (P\ \mathsf{says}\ \varphi \supset Q\ \mathsf{says}\ \varphi)}$$

$$\textit{Quoting}\ \frac{}{P \mid Q\ \mathsf{says}\ \varphi \equiv P\ \mathsf{says}\ Q\ \mathsf{says}\ \varphi}$$

$$\textit{\&Says}\ \frac{}{P\&Q\ \mathsf{says}\ \varphi \equiv P\ \mathsf{says}\ \varphi \wedge Q\ \mathsf{says}\ \varphi}$$

$$\textit{Idempotency of} \Rightarrow\ \frac{}{P \Rightarrow P} \qquad \textit{Monotonicity of} \mid\ \frac{P' \Rightarrow P \quad Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$$

$$\textit{Associativity of} \mid\ \frac{P \mid (Q \mid R)\ \mathsf{says}\ \varphi}{(P \mid Q) \mid R\ \mathsf{says}\ \varphi}$$

$$P\ \mathsf{controls}\ \varphi \stackrel{\text{def}}{=} (P\ \mathsf{says}\ \varphi) \supset \varphi$$

$$P\ \ \mathsf{reps}\ Q\ \mathsf{on}\ \varphi \stackrel{\text{def}}{=} P \mid Q\ \mathsf{says}\ \varphi \supset Q\ \mathsf{says}\ \varphi$$

Fig. 2: Core Inference Rules

$$\textit{Quoting (1)}\ \frac{P \mid Q\ \mathsf{says}\ \varphi}{P\ \mathsf{says}\ Q\ \mathsf{says}\ \varphi} \qquad \textit{Quoting (2)}\ \frac{P\ \mathsf{says}\ Q\ \mathsf{says}\ \varphi}{P \mid Q\ \mathsf{says}\ \varphi}$$

$$\textit{Controls}\ \frac{P\ \mathsf{controls}\ \varphi \quad P\ \mathsf{says}\ \varphi}{\varphi} \qquad \textit{Derived Speaks For}\ \frac{P \Rightarrow Q \quad P\ \mathsf{says}\ \varphi}{Q\ \mathsf{says}\ \varphi}$$

$$\textit{Reps}\ \frac{Q\ \mathsf{controls}\ \varphi \quad P\ \ \mathsf{reps}\ Q\ \mathsf{on}\ \varphi \quad P \mid Q\ \mathsf{says}\ \varphi}{\varphi}$$

$$\textit{Rep Says}\ \frac{P\ \ \mathsf{reps}\ Q\ \mathsf{on}\ \varphi \quad P \mid Q\ \mathsf{says}\ \varphi}{Q\ \mathsf{says}\ \varphi}$$

Fig. 3: Derived Rules Used in this Paper

The certification authority's key, $K_{CA}$, is installed on the computer using a trustworthy key distribution process, and the trust in the key is captured using the statement $K_{CA} \Rightarrow CA$. If we get a certificate signed using $K_{CA}$, then we would attribute the information in that certificate to $CA$. For example, using the *Derived Speaks For* rule in Figure 3 we can conclude that certificate authority $CA$ vouches for $K_{Alice}$ being Alice's public key:

$$\frac{K_{CA} \Rightarrow CA \qquad K_{CA}\ \mathsf{says}\ (K_{Alice} \Rightarrow Alice)}{CA\ \mathsf{says}\ (K_{Alice} \Rightarrow Alice).}$$

$K_{Alice} \Rightarrow Alice$ is a statement of trust on $K_{Alice}$, where all statements made by $K_{Alice}$ are attributed to Alice. However, in some situations, a principal may be trusted only on specific statements. For example, $K_{Alice}$ may be trusted on a statement requesting a transfer of a million dollars. However,

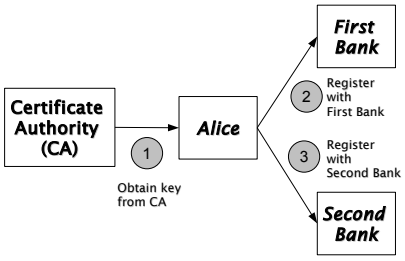


Fig. 4: Partner key management

$K_{Alice}$ may not be trusted on a statement $K_{Bob} \Rightarrow Bob$. This notion of a constrained delegation, where a principal's delegate is trusted on specific statements, is described using reps formulas. For example, if $K_{Alice}$ is trusted to be Alice's delegate on the statement $\langle transfer\ 10^6, acct_1, acct_2\rangle$, we would write: $K_{Alice}$ reps $Alice$ on $\langle transfer\ 10^6, acct_1, acct_2\rangle$.

From the semantics of reps, if we recognize $K_{Alice}$ as Alice's delegate, in effect we are saying that $K_{Alice}$ is trusted on Alice stating that she wishes a million dollars to be transferred from $acct_1$ to $acct_2$. If $K_{Alice}$ says Alice says transfer a million dollars from $acct_1$ to $acct_2$, we will conclude that Alice has made the request. Using the *Rep Says* rule in Figure 3 we can conclude:

$$\frac{K_{Alice}\ \text{reps}\ Alice\ \text{on}\ \langle transfer\ 10^6, acct_1, acct_2\rangle \quad K_{Alice} \mid Alice\ \text{says}\ \langle transfer\ 10^6, acct_1, acct_2\rangle}{Alice\ \text{says}\ \langle transfer\ 10^6, acct_1, acct_2\rangle.}$$

## 5 An Extended Example

In this section, we illustrate PKM with a hypothetical example. Suppose Alice is a cash manager who works for the Widget Corporation. Further suppose that Widget uses three banks: First, Second, and Third Bank. Suppose the three banks use different procedures for authorizing credentials, which the Widget corporate Treasurer finds acceptable. Both First and Second Banks use the PKM model, while for explanatory purposes only, assume that Third Bank uses the PKI model. Both First and Second Bank allow Alice to obtain a credential from any provider, while Third Bank requires Alice to obtain a credential from a specific certificate authority that we will refer to as (CA). Therefore, Alice obtains a certificate from CA that can be used with all the Three banks. Because First and Second Banks use PKM, Alice registers the certificate with both the banks. First and Second Bank describe their procedure for accepting certificates in a partner key practice statement (PKPS). Both First Bank and Second Bank require Alice to submit a signed PKPS along with each transaction. First Bank requires Widget to check for revocation prior to Alice sending the payment instruction. There is a mutual agreement of sender liability if Widget does not check for revocation before affixing the signature. Second Bank requires Alice

**Payment Instruction:**

1. $K_{Alice}$ says $\langle transfer\ 10^6, acct_1, acct_2 \rangle$
2. $K_{Alice}$ says $\langle First\ Bank\ PKPS, timestamp \rangle$

**Entitlement:**

1. $Alice$ controls $\langle transfer\ 10^6, acct_1, acct_2 \rangle$

**Mutually Agreed Operational Rules:**

1. $First$ controls $(K_{Alice} \Rightarrow Alice)$
2. $K_{Alice}$ says $\langle First\ Bank\ PKPS, timestamp \rangle$
   $\supset \langle K_{Alice} Validated, timestamp \rangle$
3. $\langle K_{Alice} Validated, timestamp \rangle$
   $\supset (First$ says $K_{Alice} \Rightarrow Alice)$

Fig. 5: First Bank: Payment instruction, entitlement, and operating rules

```
<pkps:pkps id = First>
 <wsp:policyattachment>
   <wsp:appliesto>
     <pkps:requester>
       <pkps:any/>
     </pkps:requester>
     <pkps:receiver>
       First
     </pkps:receiver>
   </wsp:appliesto>
   <wsp:policy>
     <wsp:all>
       <pkps:validation-model>
         <pkps:sender-no-evidence/>
       </pkps:validation-model>
     </wsp:all>
   </wsp:policy>
 </wsp:policyattachment>
</pkps:pkps>
```

Fig. 6: First Bank's PKPS

to sign an OCSP response obtained from the certificate provider, and Second Bank will validate Alice's certificate using the OCSP response. Third Bank uses the traditional PKI model, so there is no PKPS involved. Also, Third Bank uses a receiver validation model, so Third Bank will connect to the CA's OCSP responder to validate the certificates.

We will use the access-control logic (Section 3) to describe in detail the operations of the three banks for a hypothetical transaction, in which Alice requests a transfer for $\$10^6$ from Widget's account to a different account. For each bank, we provide a derived inference for justifying the bank's decision to act on the payment instruction. The proof of these derived inference rules are a direct application of the inference rules described in Section 3.3. Our objective is to primarily show the differences between PKI and PKM with respect to how the credentials are managed. We use the access-control logic to show the logical consistency of the operations and also to make the mutually agreed operating rules explicit.

**Important note:** In the hypothetical example, Alice requires an entitlement to request a transaction. The methods commonly used by banks to issue such entitlements to Alice are outside the scope of this paper. For the purpose of our illustration, we will assume that Alice has the necessary entitlement.

### 5.1 First Bank

Figure 5 contains an example payment instruction for First Bank. The payment instruction comprises two statements, (1) a statement signed using $K_{Alice}$ requesting transfer of $1 million, (2) First's PKPS (Figure 6) and timestamp signed using $K_{Alice}$. As per the mutually agreed operational rules, First has the authority for authorizing Alice to use $K_{Alice}$, and First issues such an authorization when $K_{Alice}$ is validated. According to First's PKPS, the sender is expected to validate $K_{Alice}$ prior to the transaction, and First assumes that the $K_{Alice}$ is validated appropriately when Alice signs First's PKPS with $K_{Alice}$. The following derived inference rule justifies the bank's decision to act on the payment instruction.

**Payment Instruction:**

1. $K_{Alice}$ says $\langle transfer\ 10^6, acct_1, acct_2\rangle$
2. $(K_{Alice} \mid K_{CA})$ says $\langle K_{Alice}Validated, timestamp\rangle$
3. $K_{Alice}$ says $\langle Second\ Bank\ PKPS, timestamp\rangle$

**Entitlement:**

1. $Alice$ controls $\langle transfer\ 10^6, acct_1, acct_2\rangle$

**Mutually Agreed Operational Rules**

1. $Second$ controls $K_{Alice} \Rightarrow Alice$
2. $K_{CA} \Rightarrow CA$
3. $K_{Alice}$ says $\langle Second\ Bank\ PKPS, timestamp\rangle \supset$ $CA$ controls $\langle K_{Alice}Validated, timestamp\rangle \wedge$ $K_{Alice}$ reps $K_{CA}$ on $\langle K_{Alice}Validated, timestamp\rangle$
4. $\langle K_{Alice}Validated, timestamp\rangle \supset$ $Second$ says $K_{Alice} \Rightarrow Alice$

Fig. 7: Second Bank: Payment instruction, entitlement, and operating rules

```
<pkps:pkps id=Second>
 <wsp:policyattachment>
   <wsp:appliesto>
     <pkps:requester>
       <pkps:any/>
     </pkps:requester>
     <pkps:receiver>
       Second
     </pkps:receiver>
   </wsp:appliesto>
   <wsp:policy>
     <wsp:all>
       <pkps:revocation>
         <pkps:sender-with-evidence/>
       </pkps:revocation>
     </wsp:all>
   </wsp:policy>
 </wsp:policyattachment>
</pkps:pkps>
```

Fig. 8: Second Bank's PKPS

$$First\ Bank\ \frac{\begin{array}{c}K_{Alice}\ \text{says}\ \langle transfer\ 10^6, acct_1, acct_2\rangle \\ K_{Alice}\ \text{says}\ \langle First\ Bank\ PKPS, timestamp\rangle \\ Alice\ \text{controls}\ \langle transfer\ 10^6, acct_1, acct_2\rangle \\ First\ \text{controls}\ K_{Alice} \Rightarrow Alice \\ K_{Alice}\ \text{says}\ \langle First\ Bank\ PKPS, timestamp\rangle \supset \langle K_{Alice}Validated, timestamp\rangle \\ \langle K_{Alice}Validated, timestamp\rangle \supset (First\ \text{says}\ K_{Alice} \Rightarrow Alice)\end{array}}{\langle transfer\ 10^6, acct_1, acct_2\rangle}$$

### 5.2 Second Bank

The payment instruction for Second Bank, in Figure 7, comprises three statements, (1) a statement signed using $K_{Alice}$ requesting transfer of $1 million, (2) CA's OCSP response for $K_{Alice}$ signed using $K_{Alice}$, (3) PKPS (Figure 8) and timestamp signed using $K_{Alice}$. Second Bank has authority for authorizing Alice to use $K_{Alice}$, similar to the First Bank, but uses the sender-validation-with-evidence model for validation. When $K_{Alice}$ signs Second's PKPS, both parties agree to two operating rules for validating $K_{Alice}$. First, CA has authority for validating $K_{Alice}$. Second, $K_{Alice}$ is a recognized delegate of $K_{CA}$ for relaying the OCSP response for $K_{Alice}$. The following derived inference rule justifies the bank's decision to act on the payment instruction.

$$Second\ Bank\ \frac{\begin{array}{c}K_{Alice}\ \text{says}\ \langle transfer\ 10^6, acct_1, acct_2\rangle \\ (K_{Alice} \mid K_{CA})\ \text{says}\ \langle K_{Alice}Validated, timestamp\rangle \\ K_{Alice}\ \text{says}\ \langle Second\ Bank\ PKPS, timestamp\rangle \\ Alice\ \text{controls}\ \langle transfer\ 10^6, acct_1, acct_2\rangle \\ Second\ \text{controls}\ K_{Alice} \Rightarrow Alice \\ K_{CA} \Rightarrow CA \\ K_{Alice}\ \text{says}\ \langle Second\ Bank\ PKPS, timestamp\rangle \supset \\ \{CA\ \text{controls}\ \langle K_{Alice}Validated, timestamp\rangle \wedge \\ K_{Alice}\ \text{reps}\ K_{CA}\ \text{on}\ \langle K_{Alice}Validated, timestamp\rangle\} \\ \langle K_{Alice}Validated, timestamp\rangle \supset Second\ \text{says}\ K_{Alice} \Rightarrow Alice\end{array}}{\langle transfer\ 10^6, acct_1, acct_2\rangle}$$

**Payment Instruction:**
1. $K_{Alice}$ says $\langle transfer\ 10^6, acct_1, acct_2\rangle$

**Entitlement**
1. $Alice$ controls $\langle transfer\ \$10^6, acct_1\rangle$

**Public Key Certificate**
1. $K_{CA}$ says $K_{Alice} \Rightarrow Alice$

**Trust Assumptions:**
1. $K_{CA} \Rightarrow CA$
2. $CA$ controls $K_{Alice} \Rightarrow Alice$

Fig. 9: Third Bank: Payment instruction, entitlement, certificates, and trust assumptions

### 5.3 Third Bank

The payment instruction for Third Bank, in Figure 9, is a statement signed using $K_{Alice}$ for requesting a transfer of $1 million. Third Bank believes in the jurisdiction of the CA for identifying the Key of Alice. When Third Bank receives the public key certificate for $K_{Alice}$, it validates it by connecting to CA's OCSP responder. On successful validation, Third Bank is convinced that $K_{Alice}$ belongs to Alice. For the sake of brevity, we do not describe the actual validation process in the logic. Moreover, doing so does not change the trust assumptions, more specifically does not affect Third Bank's belief in CA's authority. The following derived inference rule justifies the bank's decision to act on the payment instruction.

$$\textit{Third Bank}\quad \frac{\begin{array}{c} K_{Alice} \text{ says } \langle transfer\ \$10^6, acct_1\rangle \\ Alice \text{ controls } \langle transfer\ \$10^6, acct_1\rangle \\ K_{CA} \text{ says } K_{Alice} \Rightarrow Alice \\ K_{CA} \Rightarrow CA \\ CA \text{ controls } K_{Alice} \Rightarrow Alice \end{array}}{\langle transfer\ 10^6, acct_1, acct_2\rangle}$$

### 5.4 Analysis

The traditional PKI model is characterized by the following three statements:

| | |
|---|---|
| 1. $Key_{CA} \Rightarrow CA$ | Trust in the root key of the CA |
| 2. $CA$ controls $(Key \Rightarrow Principal)$ | CA's Jurisdiction |
| 3. $Key_{CA}$ says $(Key \Rightarrow Principal)$ | Certificate |

Users trust that the root key belongs to the CA. Trust in the key of the root CA must be established by a trustworthy key distribution process. The CA has jurisdiction over statements associating a key with a particular principal and issues PKI certificates, each of which is a statement signed by the root key that associates the key with a particular principal. The PKI model does not deal with authorizations, and authorization is considered the responsibility of the relying party (RP). Moreover, validation is also seen as the responsibility of the RP, and does not involve the user.

The PKM model is characterized by the following two statements:

1. $Bank$ controls $(Key \Rightarrow Principal)$ Bank's Jurisdiction
2. $\langle Key,\ Validated \rangle \supset Bank$ says $(Key \Rightarrow Principal)$ Bank issues authorization

The PKM model blends authentication with authorization, and Banks have the authority for authenticating and authorizing the use of credentials. The user has the freedom to obtain credentials from any provider, but the Banks reinterpret the credentials in constrained manner, which could vary between banks. In contrast to the PKI model, the validation process for the credentials is explicit and involves the user, supporting non-repudiation claims. In effect, PKPS maps the common interpretation of PKI credentials into the more constrained and controlled interpretation required by banks for high-value commercial transactions.

## 6 Related Work

There are several XML schemas for specifying web service policies and privacy policies. WS-Policy [1] is a W3C standard for specifying web service policies for security, quality of service, messaging, etc. WSPL [4] has similar motivations, but is not an accepted W3C standard. P3P enables a web site to publish its privacy practice in a machine readable format, which all browsers can read and warn their respective users if the privacy practice of a web site is incompatible with a user's personal preference [5]. Our work relates to existing XML schemas for specifying web service and privacy policies by providing a formal semantics with sound inference rules for describing policies. The benefit of our work is banks can rigorously justify acting on payment instructions based on policies and trust assumptions.

Jon Olnes [6] describes an approach that offers interoperability by using a trusted third party called the validation authority (VA). The VA is trusted by both the CAs and relying party (RP), which receives the credentials. Each VA vouches for the CAs it handles, and the RP can validate all the credentials from the CAs by connecting to a single VA. While this model provides interoperability with respect to CAs vouched for by a particular VA, it limits the RP and its customers to only those CAs. In contrast, PKM imposes no such restrictions; Banks use any CAs they want. Moreover, the PKM model reinterprets the authority of credentials in a constrained and controlled manner.

Fox and LaMacchia [7] describe an alternative to OCSP for online certificate status checking. Any method similar to OCSP that requires the RP to connect to the CA for validating the certificates, not only breaks interoperability, but also imposes a significant cost on the RP. In contrast, PKM supports a general validation model, including a sender validation model, which in conjunction with the reinterpretation of authority, scales better, provides interoperability, and reduces the cost for the RP.

Our work is related to several logical systems used for reasoning about access-control that are summarized in [8]. The access-control logic we use is based on Abadi and Plotkin's work [9], with modifications described in [10].

## 7 Conclusion

The common interpretation of PKI credentials is problematic for banks engaged in high-value commercial transactions. Partner key management (PKM),

through the use of partner key practice statements (PKPS), reinterprets PKI credentials to address the problems of scope of authority, liability, and cost inherent to high-value commercial transactions. The failure of any single high-value transaction can bring severe consequences to banks. Thus, it is essential that the policies and requirements regarding the use of credentials in high-value commercial transactions be as precise and accurate as possible. To meet this requirement, we have expressed PKI, PKPS, and PKM policies and interpretations in an access-control logic with formal semantics and sound inference rules. This enables banks and their customers to know precisely what is required of them and to justify acting on payment instructions. Our experience to date indicates that using this logic is within the capabilities of practitioners and does in fact clarify the underlying logic of credentials and their use in high-value commercial transactions.

## References


1. Vedamuthu, A.S., Orchard, D., Hirsch, F., Hondo, M., Yendluri, P., Boubez, T., Yalçinalp, Ü.: Web services policy 1.5 - framework. http://www.w3.org/TR/ws-policy/ (September 2007)
2. Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard) (June 1999)
3. Benson, G.: Portable security transaction protocol. Comput. Netw. **51**(3) (2007) 751–766
4. Anderson, A.H.: An introduction to the web services policy language (wspl). In: POLICY. (2004)
5. Cranor, L., Dobbs, B., Egelman, S., Hogben, G., Humphrey, J., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J., Schunter, M., Stampley, D.A., Wenning, R.: The platform for privacy preferences 1.1 (p3p1.1) specification. http://www.w3.org/TR/P3P11/ (November 2006)
6. Olnes, J.: DNV VA white paper: PKI interoperability by an independent, trusted validation authority. In: 5th Annual PKI R & D Workshop. (April 2006)
7. Fox, B., LaMacchia, B.A.: Online certificate status checking in financial transactions: The case for re-issuance. In: FC. (1999)
8. Abadi, M.: Logic in access control (tutorial notes). (2009) 145–165
9. Abadi, M., Burrows, M., Lampson, B., Plotkin, G.: A Calculus for Access Control in Distributed Systems. ACM Transactions on Programming Languages and Systems **15**(4) (September 1993) 706–734
10. Chin, S.K., Older, S.: Reasoning about delegation and account access in retail payment systems. In: MMM-ACNS. (2007)