Embedded Differential Access Control for Printing Variable Jobs

Helen Balinsky & Nassir Mohammad, Hewlett-Packard Laboratories, Long Down Avenue, Bristol BS34 8QZ, UK
{Helen.Balinsky, NMohammad}@hp.com

Abstract

Business case requirements have resulted in related documents that are native to their individual formats being combined into a collection called composite documents. These are used for a variety of activities such as presentations, premium/sensitive content distributions and business workflows, and may include standard types such as PDF, Word, Excel, PowerPoint Slides and other specially formatted files. Additionally, cross-organizational workflow demands have created new challenges for the management of such composite documents. These include access control to individual parts, security and privacy issues in the absence of shared trusted infrastructure, communication over popular and potentially insecure channels, auditing, print control and disposal. These challenges led to the development of Publicly Posted Composite Documents (PPCDs) [1] as a recently proposed document format that enables the secure containment and transfer of personalized document versions over insecure channels, with the access control and policies built into and carried within the document itself (as an integral part). PPCD technology is intended to act as the central pillar of document workflow scenarios, where a single document can be created, controlled, accessed, used and monitored throughout its life cycle as it participates in inter-organizational workflows. However, PPCD technology is currently missing an important property of electronic documents: the ability to be securely printed to a physical copy. The limited resources on multi-function printer (MFP) devices and the security focused structure of PPCDs makes this a challenging and non-trivial problem. Thus, in this paper we utilize the unique structure of PPCDs to describe novel mechanisms and firmware extensions on MFPs for enforcing differential access control and printing.

Introduction

The way in which documents are consumed by society is a rapidly evolving process. Traditional physical copies of literature, reports, spreadsheets, utility and bank statements, manuals, etc. have given way to paperless electronic versions that are seen as economical, convenient, environmentally friendly and efficient alternatives. However, whilst some traditional areas of printing, like photo printing are reducing, the overall trend is an upward rise in the number of professional print jobs: it is forecast that by 2021 Global production will be 1.2 Trillion pages [2]. More recently, documents which were thought of, and utilized as, unitary elements are often now combined into composite documents containing several or more natively formatted and accessible parts. These parts may be composed of individual images, Microsoft PowerPoint presentations, Microsoft Word documents, Adobe PDF files, text files, etc. Such composite documents are natural in the current climate where typical presentations, business workflows and content distributions may include several, if not all, of these formats. Furthermore, composite documents also provide users with the convenience of handling and communicating all related materials together in one easily accessible file.

As the use of composite documents has expanded, so has the potential to enable these documents to act as more than just information holders. Indeed, demand for differential and temporal access control for different parts of a document have stemmed from the utilization of these documents in multi-participant cross-organizational workflows, where each user is granted access when it is their turn to use the document. Workflow requirements also necessitate differential and dynamic access control to each individual part. Ideally, a creator of a secure composite document wants to specify different access rights for each user and for each part at different times and stages in the workflow. These privileges may include the right to verify authenticity of some parts to ensure overall composite document authenticity, read or read/write access to other parts.

In instances where a composite document is to be utilized within a trusted domain it is possible to satisfy the privacy and security rights of each individual by providing access through a central authority. However, in a highly interconnected world of specialist organizations, documents are often required to cross boundaries in complex workflows that involve participants in different geographical and business environments, e.g. partners, consultants, collaborators, clients. For each organization or security environment documents are required to be exported and re-imported after being updated. The management of secure access rights to view and edit these documents poses a significant challenge to organizations wishing to balance productivity and security in a digital environment. From a security perspective, the provision of a mechanism to allow the creation, management, posting and differential access of composite documents is a significant technical challenge to existing solutions [3][4].

The issues relating to access control of composite documents was raised and addressed in the paper [1], where a detailed description of a new document format: 'Publicly Posted Composite Documents' (PPCDs) was presented. Issues relating to the secure access control, authentication, efficiency of a document, encapsulation of different traditionally formatted documents for user convenience and propagation through inter-organizational workflows were addressed. The paper [5] also solved an important requirement of users being able to modify existing content in the PPCD by scanning documents at the device. However, complementary workflow challenges remain to be tackled in order to utilize fully PPCDs as regular documents and to expand their functionality.

In many scenarios, a key property required of a document is the ability to print a physical copy. Indeed, many business compliance rules still require paper copies of important documents, such as employee contracts, to be retained for record keeping. Applications for mortgages, jobs or contractual agreements often require the person to print a paper copy, sign, scan and email the electronic version to the specified recipient. Copies of highly secure and sensitive documents such as passports and driving licenses also require printing, as well as premium documents in high quality print distribution scenarios. Documents are also printed for convenience such as for reading, sharing, portability and circulation. Furthermore, printing of electronic files is not just restricted to paper outputs, but also to the printing of objects using 3D-printing devices. PPCDs are also not immune to this requirement, and one of the key properties PPCDs need to have is the ability of users to print securely them to obtain their personalized document version i.e. all or some parts to which access is granted.

In this paper, we detail how the structure of PPCDs can be securely processed inside HP Multi-function Printer (MFP) devices so that they can be securely printed like any other regular document. The job receiving, storage, data authenticity check, access control enforcement, retrieval of private keys, decryption, sensitive data handling and personalized printing are all carried out in real time inside the device itself. Unlike other common document formats, the security and differential access focused structure of PPCDs makes this a challenging and non-trivial task to implement through firmware extensions on MFPs with limited resources.

The rest of the paper is organized as follows. In the next section, we describe the security issues and challenges of printing PPCDs. For the reader's convenience, we then recall the original PPCD technology and structure through a sequence of document access. Then we provide an overview of the architecture of our solution that enable PPCDs to be securely printed, followed by a section on the technical implementation details. We then detail the benefits of our solution and real time printing of PPCDs in office environments. Finally, we conclude the paper and provide our acknowledgements and references.

Security and Challenges of Printing PPCDs

We are not aware of any existing solutions that allow *composite* documents with cryptographically built-in access control to be securely sent to and printed on an MFP device. Currently, the only method of printing a PPCD is to receive it on a personal device, extract (decrypt) individual components and then print each component using its own application or dedicated PPCD Authoring Environment desktop application [6]. To protect sensitive contents in this scenario, users must ensure that the decrypted document-parts are re-encrypted or a secure communication channel is used when sending to print [7]. Stored jobs must also be encrypted in the device whilst awaiting the user's access. To remove the complexity and security vulnerabilities arising from the need to access a PPCD on two devices just to produce a printed copy of some parts, this paper proposes a dedicated print solution, where the PPCD formatted document can be processed directly at an MFP device.

The proposed solution provides the full cycle protection for the sensitive data locked within a PPCD: whilst transitioned to residing inside a printing device. The new solution improves security of data handling whilst greatly simplifying document workflows by eliminating unnecessary data decryption and re-encryption outside of a printing device and reducing the number of steps required to print some or all parts of a PPCD. Enabling PPCDs to be printed on a standalone device also solves several workflow problems. The encrypted format means received documents do not have to be pin protected to prevent printing without the presence of the pin holder. They can be left on the device storage and accessed anytime by a user simply providing their single private access information to print all or some individual document-parts to which he has access granted. Additionally, since PPCDs also provide differential access, documents can be securely stored on a single shared device so that users can print their personalized version of the document, e.g. a payslip.

In order to enable printing of PPCDs, several challenges and security problems need to be addressed and solved which we discuss in the following section.

Problems and Challenges

Many challenges are encountered when trying to enable PPCDs to be printed on an MFP. These are further exacerbated by the targeted enterprise level MFP devices having limited resources and power due to business cost constraints. The security driven nature of the problem also means that every aspect of a solution must ensure a tightly secure system that does not compromise sensitive information.

A device must be enabled to accept the specially formatted and encrypted jobs without automatically submitting them to the printing pipeline within the device: awaiting a user to provide the decryption keys. Instead, such documents need to be diverted from the standard path of a print job and stored in a dedicated/secure location on the device, accessible through authentication. . Furthermore, a solution enabling this should not interfere with other types of unrelated print jobs (standard or special) that may also be submitted to the device.

Further challenges include users being provisioned with a way to navigate the menus on a device's control panel to access all or a sub-selection of currently stored PPCDs - depending on the preferred deployment mode. They may also need to be provided with facilities to access their on-line identity or engage their private key (required to access their documents) whilst at the device, and be able to select a PPCD for printing.

PPCD access control and decryption, in device, also needs to be implemented without the need of an external server, as the access control is an integral part of the document itself. This means that solutions should be incorporated into a single standalone device, without any associated server, networking or communication overhead, thus making it a convenient, economic and secure solution. Furthermore, in many countries, especially in Europe, outsourcing of the decryption and/or signature from the MFP may invalidate workflow compliance.

Any firmware extension solutions that enable printing on the device must also ensure that the user's private key brought or engaged with the device is handled securely. In some cases, it may even be necessary for a user to provide their private key and login credentials to view the documents in a secure partition of the device. Furthermore, any subsequent keys that are decrypted for accessing the document and its parts must be handled securely, i.e. accessed only when needed, and discarded immediately after being used. The differential access control on device must also ensure that any decrypted personalized document versions are sent securely to the print pipeline and discarded immediately after being processed.

Before proceeding to present our solution, let us recall in the next section the main components of a PPCD and the steps to follow in order to access the document by a workflow participant: the steps to be replicated within an MFP device.

Overview of PPCD by Access Procedure

PPCDs were introduced in [1] as a new composite document format to support the secure creation and management of composite documents, with variably formatted and differently sensitive data in multi-user document workflows. The motivation for this novel format over existing solutions, e.g. [8] stems from the need for sensitive document workflows not necessarily contained within one secure environment and at the same time the absence of a trusted central authority for document access. Instead, PPCD technology allows a single composition of individual traditional documents to be differentially accessed by many users without concerns over organizational boundaries and low-security communication channels

A PPCD is a SQLite serialization of multiple differently formatted parts, traditional office documents files and/or fragments-with built-in fine-grained access control and enforced variable workflow order. These parts may each be an image, a spreadsheet, word document, music file, video file, etc. The access for workflow participants is provided through known public keys of participants and/or Public Key Infrastructure (PKI). Prior to creating a PPCD document, the document master is required to know the public keys of the involved participants. In deployment modes where this requirement may be difficult to satisfy, Identity Based Encryption technique could be deployed [9]. These methods of selecting workflow participants allow the assignment of access rights to each individual part and the creation of flexible workflows according to business needs. Furthermore, this ensures that only specified and authorized users can access their personalized versions from the same document version according to individual workflow roles and access rights assigned by the PPCD master.

Access Initiated at the entry-table

PPCDs have been designed to contain any mixture of confidential and publicly accessible data. The sensitive data is encrypted and inaccessible with the entry-table made a starting point to PPCD access by workflow participants. The entry-table consists of randomized rows with at least one row present and corresponding to each workflow participant. Without a corresponding record in the PPCD entry-table a workflow participant cannot access the document. The randomization ensures that no correlations exist between the access orders required by a document workflow. Additionally, dummy rows may be added as a form of padding to conceal the actual number of participants in sensitive workflows. The workflow participant's private decryption key is required to identify and extract information from the corresponding record in the entry-table. Each individual record includes the following four separate fields (Table 1):

1. The cipher text of the user's access symmetric key K, which is encrypted using the user's public encryption key P: Enc$_P$(K).
2. A plaintext "Magic String" (i.e. any string of characters).
3. The cipher text of the same "Magic String", encrypted using symmetric key K: Enc$_K$(Magic String).
4. The cipher text of the encrypted key-map entry name or ID in the document serialization where it is encrypted using key K: Enc$_K$(key-map).

Table 1: Workflow Participant row of PPCD entry-table

Enc$_P$(K)	Plaintext Magic String	Enc$_K$(magic string)	Enc$_K$(*key-map*)

This table illustrates the column fields found in the entry-table where, from left to right, we have the workflow participant's symmetric key K encrypted using their public key, the plaintext magic string, the plaintext magic string encrypted using key K, and the participant's key-map name/ID encrypted using K

Table 2: Key-map table example row

Part ID	Verification Key	Part Name	Decryption key	Encryption Key	Signature Key

This table illustrates the column fields found in the workflow participants key-map table. From left to right, each row will have, subject to the access granted, the Part ID number, a signature verification key used to verify the corresponding part signature in the parts table, the name of the part, the part decryption key, the part encryption key and the part signature key. Note that all the fields, except the Part ID, are encrypted using the symmetric key K obtained from the entry-table.

Table 3: A Parts table example row

Part ID	Part Data	Part Signature

This table shows the column fields found in the parts table which contains the actual encrypted and signed document contents. From left to right we have the unique Part ID, the encrypted document (e.g. PDF) data and the signature of the encrypted part. The signature verification key (and decryption key where applicable) is provided in the workflow participant's key-map table.

Key-map Entry Recovery

The next step in accessing a PPCD is the recovery of the user's key-map entry. These contain individual subsets of access keys for every document part, corresponding to the access granted to each workflow participant. Each individual key-map entry is encrypted using the corresponding participant's symmetric key, K (Table 2), recovered from the entry-table. The identifier to this table is encrypted using K and recovered from the user's entry-table row. This ensures that the key-map entry is only accessible by its intended workflow participant.

Once the user's key-map is recovered and decrypted the individual subset of access keys for each part in the parts table (Table 3) are retrieved. As the first step, authenticity of each part is verified using the signature verification key provided for each part irrespective of access granted. Furthermore, for each part with at least read access granted the corresponding decryption key is used to extract the part contents. For parts with modification right (read/write) the encryption and signature keys allow the participant to legitimately modify or replace the part contents without infringing the overall document authenticity.

Verification of Authenticity and Document Access

Fig. 1 illustrates the process the device's firmware needs to follow on behalf of a workflow participant, to verify PPCD authenticity prior to performing decryption to recover their personalized content (access parts with read access). Since a PPCD is a composition/serialization, containing various tables representing access, verification and content-parts of the document, the process begins with verification of the entry-table, key-map and signature tables (1.a), which are signed by the document master and whose public key certificate could be stored in a master certificate table if no trusted certificate is known for them. All table signatures are stored in the signatures table (which is itself signed) and must be verified to ensure document authenticity. Note that the document creator's certificate is verifiable as it is signed by a trusted Certificate Authority (CA). If any signature verification fails (1.b), it indicates that the PPCD has undergone unauthorized changes, and a user may be prompted as to whether they want to continue. In alternative deployments a user may be prompted to provide a trusted master's certificate through a USB device or fetched through on-line services. In another alternative development, when authenticity of a PPCD is not questionable, a user may waive the authenticity check and proceed directly to content recovery.

Once PPCD authenticity has been successfully verified or waived, firmware on behalf of a workflow participant proceeds to decrypt the document by following standard PPCD protocol. First, the workflow participant's private key must be accessed. Here there are a few methods available. A participant's private key can be accessed through a locally provided smart card, if access is provided on the printing device. It can also be accessed from a USB device (e.g. ActiveIdentity), an LDAP repository or a trusted on-line/cloud service. In these cases the encrypted and wrapped key are brought to the device in a secure format, e.g. PKCS#12, where a user might be prompted to provide an access password or PIN.

Once a program successfully navigates the entry-table (1.c) and recovers the necessary part keys from the user's key-map entry (1.f), all the part signatures in the parts table can be verified (1.g). In cases where a workflow participant has only verification access granted, the processing stops here. For example, such access could be granted to a printing device, enabling immediate verification of a received PPCD. If a corrupted PPCD is received, the device can immediately discard it and communicate to the sender/administrator (via e-mail) requiring re-submission over the same or different channels.

In deployments where authenticity checks are delegated to the printer, if authenticity of stored data can be guaranteed, then on user's access the document verification stage can be skipped, further reducing access time, proceeding directly to decryption and printing. In other instances where the workflow participant has at least Read Only access to at least one part, the part symmetric decryption keys can be used to decrypt (1.i) and recover each part with at least Read access in its clear-text format and have them submitted to the printing pipeline (1.j).

It is important to accentuate that the access, and hence the printing solution, allows for differential printing of the PPCD to provide the portion of the document commensurate with the granted access rights. Document users may also be shown a selection of printable parts of the PPCD and print only a sub-selection using the device's control panel. This may be administered from a single document residing on a single easily accessible MFP device for all users to access and use in a serial fashion (e.g. for printing personal pay-slips). On the other hand, the same document may also be printed by multiple users at their workflow step to give their personalized version. Furthermore, supplementary parts could also be provided for different classes of users, e.g. new employees could receive explanations regarding their pay-slip, while general cover letters are provided for all others.

Printing Solution Overview

The printing solution for PPCD is designed using the HP Open Extensibility Platform for devices (OXPd) and embedded Solution Development Kits (SDKs) [10] available on HP MFP's running Windows CE. The challenges in printing PPCDs are each addressed by different modules to create a unified solution for controllable and secure printing:

- Dedicated Input/Output filter module for PPCD jobs that are identified on arrival and stored locally awaiting user's access. (Note that an incorrectly formatted PPCD job maybe unprocessed by the filter and entered into the printing pipeline as an assumed regular print job. However, this will result in incomprehensible printed documents since printable jobs need to be formatted to a printer recognizable language, e.g. PDF for correct printing.)
- Device's control panel navigation and UI module: A separate module installed on the device is used to provide users with a navigation control where they can navigate to stored documents, select a method for private key access and signature verification certificates.
- User private key access module: A module for accessing a user's private key or engaging it with the device.
- PPCD authentication and access module: follows the PPCD authentication and access procedures, including identification of the participant's row in the document entry-table, decryption and recovery of individual document parts.
- Submission module: submits the clear-text document parts to the printing pipeline.
- Sanitation module: securely removes all sensitive data after the workflow participant completes his step or when any error is encountered by the device during document processing.

The following subsections describe these solution components in more detail with reference to the schema in Fig. 1.

Print Job Preparation

Unlike other printing formats, PPCD jobs are encrypted and cannot be directly submitted to the printing pipeline. In order to send a PPCD to a device and have it processed correctly PPCDs are wrapped into a specific PJL wrap (Fig. 2.a); ready to be received by a device with a dedicated filter module installed.

Receiving Jobs at a MFP

A print job can be sent to a standard 9100 Web Services for Devices (WSD) listener port (Fig 2.b) of the printing device, where the job is accepted by active listeners. Since many different formats can be accepted (PCL5, 6, PostScript, PDF, etc.) printing devices contain Input Output (IO) filters which sort the input jobs according to their formatting and route them accordingly. These filters are designed to execute only if format specific tags are found, and otherwise to leave the data unaltered for another process to handle. Once recognized, a standard print job is submitted to the printing pipeline. In order to handle PPCD jobs, a dedicated filter called PPCD-filter (Fig. 2.c) has been developed for an MFP device. Similar to other filters on a printer, it scans for PPCD specific commands. Thus, when a PPCD wrapped with such commands and supplemental job information (e.g. name of the file, size, etc.) is sent to a device (Fig 2.a), the filter recognizes the job and continues to processes it accordingly (as opposed to submitting

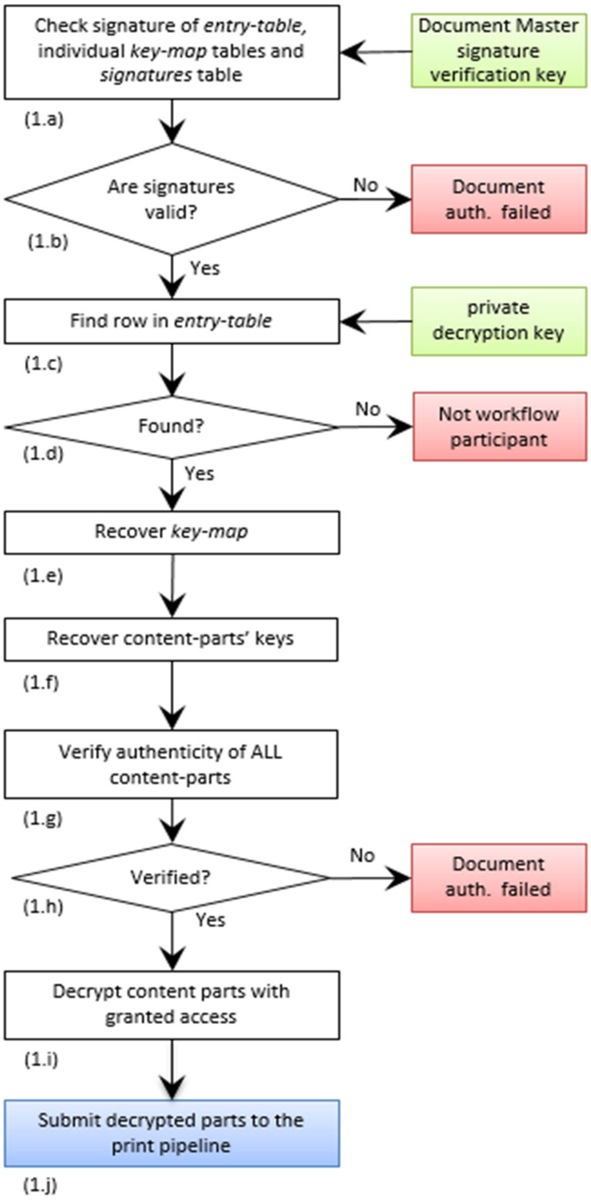

Check signature of *entry-table*, individual *key-map* tables and *signatures* table
Document Master signature verification key
(1.a)
Are signatures valid?
No
Document auth. failed
(1.b)
Yes
Find row in *entry-table*
private decryption key
(1.c)
Found?
No
Not workflow participant
(1.d)
Yes
Recover *key-map*
(1.e)
Recover content-parts' keys
(1.f)
Verify authenticity of ALL content-parts
(1.g)
Verified?
No
Document auth. failed
(1.h)
Yes
Decrypt content parts with granted access
(1.i)
Submit decrypted parts to the print pipeline
(1.j)

Fig. 1: PPCD access procedure to be followed by MFP

it to the printing pipeline on reception). Alternatively, extensions of document retrieval could include a PPCD formatted document (in its original encrypted formatting) being pulled down from a server (Fig. 2.n), repository/service such as HP Flow CM Professional [11] (Fig. 2.o), from removable media (local USB drive) or a remote drive (Fig. 2.p).

Gathering User Credentials

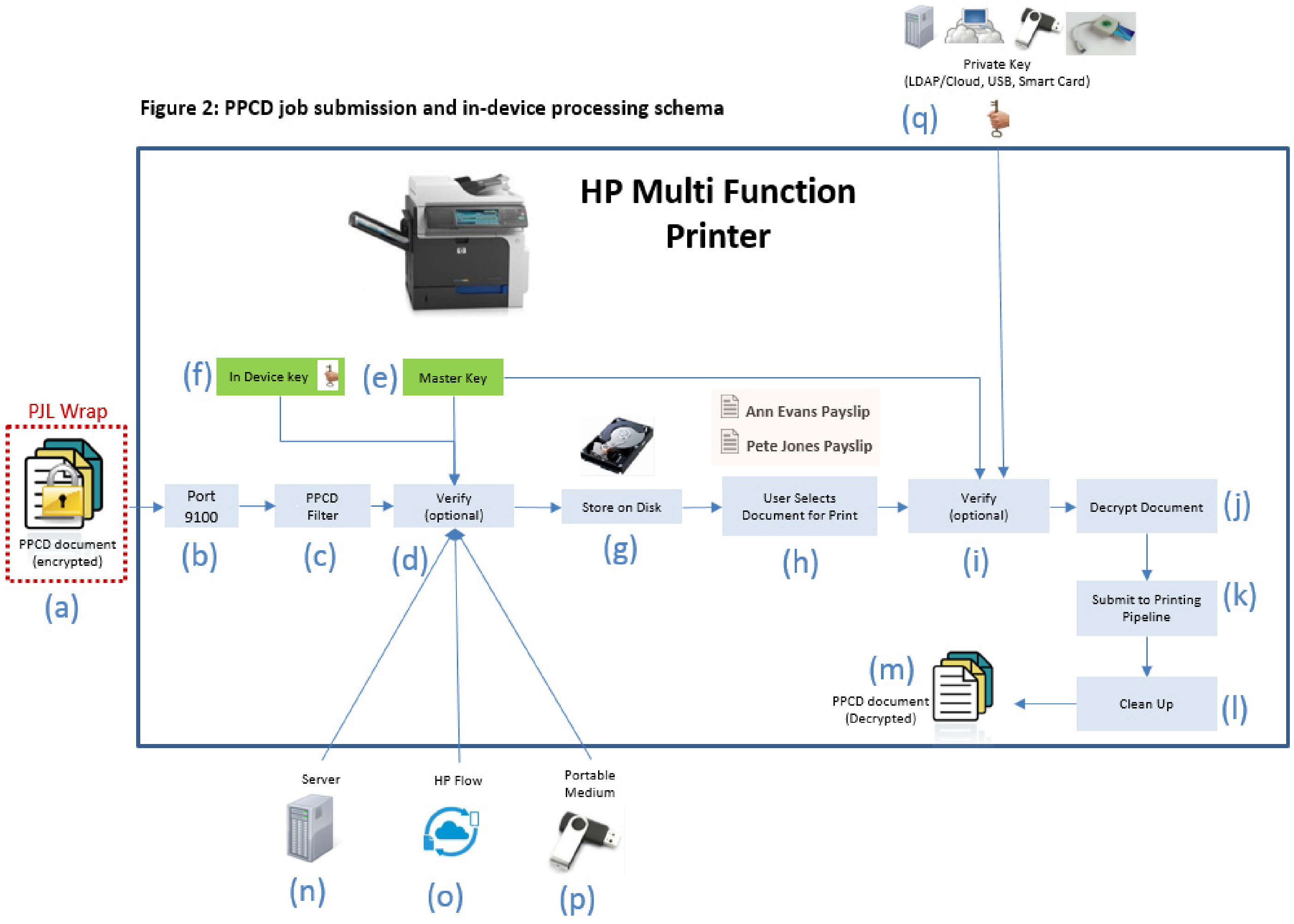
The mandatory objects required to print PPCD jobs by a participant include their corresponding private key, which must be present at the device for decrypting the document. The private key access module obtains this in a variety of ways (Fig. 2.q). Where the device is a workflow participant the keys can be accessed from within the device secure storage using OXPd API. However, where users need to bring their private key to the device, several methods can be used. In one scenario, the front panel of the printing device may offer a dedicated button to access stored PPCD-formatted print jobs and a user may be prompted to select a source for their private key. This could be fetched from a LDAP service (a choice of known and trusted LDAP services or other cloud-based identity services may follow) or a user may be allowed to select an alternative service. The private key could also be brought to the device directly by a user where it can be accessed from a portable memory device e.g. USB stick storage device or Smartcard.

Verification of Document Authenticity by a Firmware Solution Workflow Participant

If PPCD authentication is required and performed by a printer on job arrival, then the authentication module can immediately perform verification (Fig. 2.d) prior to saving onto a secure location on the devices disk drive (Fig. 2.g). This immediate verification requires a trusted document master's signature verification certificate (Fig. 2.e). The certificate can either be read from device's secure key storage or retrieved from a print job itself, where it needs to be signed by trusted CA. Additionally, the MFP device(s) also needs to have a private (and public) key (Fig. 2.f) so that they can be made workflow participants who have access to all the part verification keys. Otherwise, document validation must be deferred to a later stage, and can only be performed during user access (Fig. 2.h, i). Following the flow chart of Fig. 1, if either of the following is true: a) trusted certificate cannot be found or trust to a provided certificate chain cannot be established (does not contain trusted root CA), b) signature verification fails on either steps of the flow chart Fig. 1.b and h; then the received document is automatically discarded. A notification may be issued to the job sender or device administrator and logs can be generated and emailed using the device's built in email functionality to create a tightly monitored system where PPCDs are only accepted from authorized parties.

Accessing a Stored PPCD job in an MFP

Figure 2: PPCD job submission and in-device processing schema



PPCDs may be locally stored on a device or retrieved remotely by the MFP for job processing. The document may be pulled down from a server, which is especially beneficial for devices with solid state storage instead of disk drives. The solution can also be extended to include multiple printers as part of a network where a centralized server holds a PPCD file and users can access this file from any printer within the environment. The selected printer would then contact the server and use the workflow participant's credentials to print the granted accessible parts of the PPCD. Such solutions provide convenience to workflow participants who can access the same document across different devices.

The directory listing of PPCD files on the device (Fig. 2.h) can be provide through a UI module and is open to all users to view or protected behind successful authentication. If the user decides not to proceed with a print job, the directory listing is then automatically closed within a short pre-set period of time and any user credentials discarded. In other deployments, without loss of generality/deterioration in security, this sequence of operations could be reversed: selection of a job to print prompts users to provide their private key. This could be particularly beneficial if encrypted and unencrypted jobs are to be stored in the same location on the device. In yet another embodiment, even further isolation can be achieved by having each job placed into a dedicated directory derived from user login information.

Decryption, Print Pipeline Submission and Clean Up

Once a user's private decryption key is safely accessed and print initiated by the user on a selected file, PPCD access functions are executed by the access module together with the signature verification functions (Fig. 2.i) if applicable. If any of these verifications fail or the participant does not have document access on the device, the job processing is terminated immediately and all sensitive data is discarded by the sanitation module. Otherwise, the document is decrypted by the decryption module (Fig. 2.j) and the user accessible parts stored in a temporary directory (created for the current user session) with exclusive access to the printer firmware. The parts are then automatically submitted to the printing pipeline by the submission module (Fig. 2.k) using OXPd API so that the clear-text parts cannot be intercepted during transmission. The decrypted contents and any keys in the temporary directory are then securely discarded (Fig. 2.l) to ensure that all sensitive data is disposed of.

In some scenarios, the user may be notified by a message displayed on the device's Control Panel confirming the print job has been completed. The user may then be prompted to print more job(s) from the directory listing the PPCDs using the same credentials. However, if no jobs are required, the user can sign out or will be automatically signed off after a preset period of time.

Implementation Details

The implementation details are given here by expanding upon the solution overview provided in the previous section, with specific reference to the solution modules. The details are specific to HP MFP technology, however, we also utilize open source libraries to allow authenticated users to differentially print PPCDs. We also note here that for security reasons, enterprise level secure HP MFP devices require any firmware extensions to be signed using a valid key. Otherwise, any such extensions are rejected on installation by the devices.

Wrapping, Sending and Receiving PPCDs

Sending to a standard print port 9100

As any standard job to be processed by a printing device, a PPCD needs to be wrapped into PJL commands that are recognizable by the device. This is done in the special PPCD Authoring Environment [6] or using purposely-built tools (drivers), which automatically wraps and submits PPCD jobs, like any other print jobs.

Any individual-component parts in their native, non-printer-recognizable formats are replaced or augmented by their printable alternatives by either PPCD Authoring Environment or PPCD drivers, which make use of the corresponding native applications drivers. For example, a printer non-recognizable Microsoft Word file could be converted into the corresponding printer-native PCL format or recognizable PDF format, using application's print driver. Depending on the subsequent use of the PPCD-formatted job, the original parts could be retained or discarded.

To distinguish PPCD-formatted jobs, from other device acceptable jobs such as PCL, PS, PDF, text or raw images, a dedicated PJL command, "@PJL PPCD' is introduced. This command should not be recognizable by any other installed Input Output Filter. The following illustrates an example of a PPCD job wrapped into PJL:

```
    @PJL PPCD
    @PJL JOB NAME = "PPCD JOB"<CR><LF>
    @PJL PPCD FORMAT:BINARY SIZE=int NAME =
pathname <CR><LF><binary data>
    @PJL EOJ NAME = "End of PPCD Job"<CR><LF>
```

Figure 3: Example of PPCD wrapped by special PJL commands

Here the '@PJL PPCD' signifies the keyword that the corresponding filter should pick up. '@PJL JOB NAME' states the name of the job. The size of the actual job being wrapped is provided by 'SIZE' along with the actual binary data. Finally, the wrap is ended with the command '@PJL EOJ Name' to indicate the end of the job to PJL parsers.

Receiving at the port 9100 inside printing device

To process a print job, a dedicated Input/Output Filter needs to be added to the input port listener in a device. In the presence of other formatted print jobs the filter needs to:

- ensure that a PPCD job is rejected by any Input/Output Filter, installed in a device for handling other jobs;
- identify PPCD jobs by the dedicated filter and remove them from input stream for further processing;
- allow any standard job to pass through PPCD filter uninterrupted, so the new filter does not interfere with other jobs, whose filters appear later in the queue of installed filters.

The dedicated PPCD Input/Output filter module, written in C/C++, is added to the sequence of the filters by the input listener. The PPCD-filter expects the dedicated PJL command '@PJL PPCD'. When the tag is detected, the PPCD Input/Output filter parses the incoming data stream and the job is removed from the input stream. If the tag cannot be detected, the input bytes are output back into the stream unaltered for the subsequent filter/process to handle.

Using the value of SIZE parameter from '@PJL PPCD FORMAT:BINARY' command the filter determines the size of an incoming PPCD job, which is copied from the input stream to a designated storage location. The storage location is generally a temporary/customer partition of the device's internal storage, configurable by a device administrator. The dedicated location may also be access-controlled depending on security requirements. The NAME attribute is used to determine the name of the file for a PPCD job to be stored. If a file with the same name already exists in the specified location on the disk, then a simple version control is used to avoid overwriting the previously stored job, e.g. concatenation of a timestamp to the filename. Depending on the deployment scenario, received jobs that are not accessed for a specified period of time are automatically discarded.

Job authenticity check on arrival

A device's PPCD-filter receiving a PPCD-job can be enabled to check the documents authenticity on arrival using the authentication module. This can be done when a printer has its own pair of public/private keys and is made a workflow participant by a document creator. This is so that the document signatures can be verified before saving to the secure storage location, with documents that have had unauthorized changes being discarded. The verification procedure may also be repeated when a user later selects a document for printing.

Authentication and Retrieval of User's Private Key

The PPCDs stored in a device can be protected from unauthorized access by enforced access control for allocated storage. In the current implementation, upon attempting to navigate to the protected storage, a user is prompted to authenticate against a preset LDAP server. Upon successful authentication the user's Private Key (in the current prototype RSA, 256 bits key) is fetched and delivered to the device by the private key access module. To ensure that the confidentiality of the user's key is not compromised, keys are both stored and transferred in an encrypted form, e.g. PKCS#12 format. The private key is neither accessed, nor decrypted until the actual printing of the job and is discarded immediately after.

This process is implemented by extending C# OXPd technology, so that the authentication function utilizes unmanaged calls to a C++ Open LDAP DLL. Here, an LDAP connection is initialized with the server by supplying a distinguished name and some type of authentication credential, such as a password (retrieved from the credentials input by a user at the device). Upon a successful authentication, the user's private key certificate is automatically retrieved. In future the private key could also be fetched from a Cloud based identity or smart card readers could also be connected and used so that users never even have to expose their private key to another device.

Processing a Job

In the current prototype successful authentication results in the OXPd managed code UI module displaying the list of stored PPCD jobs. When the user selects a PPCD-job, the list of individual content-parts (which are in printer recognizable format, e.g. PS, PCL, PDF, text, etc.) with access granted to the user is displayed. The user is prompted to select all or some content-parts for printing. The printing is initiated by pressing the device's Start Button (Green Button).

Decryption and submission to Print Pipeline

Selected content-parts are decrypted and submitted to the printing pipeline. Since a PPCD is a SQLite database, this process involves using the open source SQLite C++ library to read the data, whose unmanaged source functions are imported into C# (OXPd). All cryptography objects and functions are carried out using BouncyCastle (C#) [12].

Once the document parts have been sent to the printing pipeline all sensitive and temporary files are securely discarded by the sanitation module, using OXPd API to specify the paths and files to be securely deleted. These include the private key, user password if input, the decrypted parts to be printed and any folders holding files to be emailed or processed after printing. We stress here that any sensitive data brought to the trusted device, or any content that is decrypted and temporarily stored on the device, is in a secure partition that is only accessible through OXPd API. Depending on a deployment scenario, the processed print job is either discarded or left on the device for other users' access.

Solution Validation

The solution we have presented in this paper allows users to print PPCDs securely on an MFP device without concerning themselves with additional computers, data security and authenticity, or communication overhead. The document access and decryption are all carried out inside the device without reference to any external resources. The solution has been installed and tested in select real office environments around the world with HP MFPs, and has enabled users to print PPCDs quickly and efficiently – just like the printing of any other regular document. From a user perspective, document users are only required access to their private key to process multiple documents securely on a standalone device for printing. Hence, this is an efficient, secure and user-friendly solution for printing PPCDs. Document creators can also now send a single document to an MFP with multiple users able to engage the single standalone device to print their personalized version of the document. With the solution in a stable working state, it will be useful in future to gather user feedback on the user interface for improvement.

Conclusion

We have shown how an MFP device can be enabled to receive and authenticate PPCD jobs and to store them securely on the device's internal storage. Furthermore, functionality has been embedded inside a device so that authenticated users can use their private key to navigate the security focused structure of a PPCD to yield personalized prints. Workflow participants can import their private keys to the device from on-line services such as LDAP/Cloud servers, access them from local portable storage medium or Smartcards. The decryption of the document is done using the BouncyCastle Library and carried out inside the device itself. The decrypted plaintext content-parts are then submitted to the printing pipeline to yield physical copies of the parts, and any sensitive data is discarded to reduce the risk of exposing any confidential data. The cluster of mechanisms and firmware extensions thus enable PPCDs to be printed on (HP) MFPs. This is a complementary solution to our scan-to-PPCD solution in [5], with which we note that the type of access rights given to content-parts effectively determines the processes that can be carried out at the device. i.e. read only access to a content-part given to a user corresponds to being able to print the part, whilst read-write access enables the user to securely add encrypted and signed scans to the PPCD.

In future, we would like to increase MFP functionality to make the devices into first class workflow participants that can engage with users as standalone devices. With devices enabled to print PPCDs, we now want to explore opportunities to utilize their unique structure to guide users through automated workflows that can be wholly completed on a device without referring to a personal computer (e.g. securely adding scans certified at the device and pushing entire documents to the next workflow participant). We also want to explore ways for document creators to manage documents after creation which will be applicable in scenarios involving limited printing of premium and sensitive content, e.g. restricting printing to a particular set of devices. These solutions have garnered significant interest from governmental and financial institutions.

Acknowledgment

We would like to thank Andrew Spencer for his vital and beneficial help in understanding and leveraging HP OXPd technology to enable PPCD printing.

References

[1] Balinsky, H.Y. and Simske, S.J. Differential Access for Publicly-Posted Composite Documents with Multiple Workflow Participants, ACM Symposium on Document Engineering, pp.10, 2010, Manchester, UK

[2] The future of print: digital print trends in a graphic overview, http://www.xerox.com/digital-printing/future-of-print/enus.html

[3] Wainer, J., Barthelmess, P., Kumar, A., W-RBAC-a workflow security model incorporating controlled overriding of constraints, Int J. Coop. Inf. Syst. 12 (4) (2003) 455-485

[4] Sandhu, R., Coyne, E., Feinstein, H. and Youman, C., Role-based access control models, IEEE Computer, v 29 (2), Feb 1996, pages 38-47

[5] Balinsky, H.Y. and Mohammad, N. NIP & Digital Fabrication Conference, 2014 International Conference on Digital Printing Technologies, pp. 366-371(6)

[6] Balinsky, H and Perez D., Handling Environment for PPCDs, Data-Driven Process Discovery and Analysis, LNBIP, Volume 203, 2015, pp 48-64.

[7] Secure Encrypted Print with HP UPD, http://h20331.www2.hp.com/Hpsub/downloads/UPD%205%203%20brief.pdf

[8] FileOpen Document Rights Management http://www.fileopen.com/products/rightsmanager/

[9] Balinsky, H.Y., Chen, L., and Simske, S.J. Publicly posted composite documents with identity based encryption, DocEng '11, Proceedings of the 11th ACM symposium on Document Engineering, Pages 239-248

[10] OXPd and Embedded SDK overview of proprietary software, http://h71028.www7.hp.com/enterprise/downloads/OXP_brief_4AA0-4557ENW_final.pdf.

[11] HP Flow CM, https://www.hpflowcm.com/

[12] Bouncy Castle Library, https://www.bouncycastle.org/M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1981.

Authors Biographies

Dr. Nassir Mohammad is currently a Research Scientist in the Printing and Content Delivery Lab at HP Labs, UK. He works in the Automatic Policy Enforcement eXchange Project with interests in information processing technologies; in particular, document workflows, security, data leak prevention, data mining and applications. He completed his PhD in Computer Vision at Cardiff University and HP Labs, and holds a MSc. in Mathematics and Computing from Swansea University, UK and a BSc in Mathematics from Cardiff University.

Dr. Helen Balinsky is a Principal Research Scientist from HP Labs, UK and is the Technical Lead for Automatic Policy Enforcement eXchange Project in Printing and Content Delivery Laboratory. Her research interests include: document security within an organization and in cross-organizational workflows, Data Leak Prevention, data-mining and security applications. Helen received her Master and PhD degrees in Applied and Industrial Mathematics from Technion –Israel Institute of Technology. Helen is a Fellow of The Institute of Mathematics and its Applications (IMA).