A Privacy-Aware Service Discovery Middleware for Pervasive Environments

Roberto Speicys Cardoso, Pierre-Guillaume Raverdy, and Valérie Issarny

INRIA Rocquencourt 78153 Le Chesnay, France
{first-name.last_name}@inria.fr

Abstract. Pervasive environments are composed of devices with particular hardware characteristics, running various software and connected to diverse networks. In such environments, heterogeneous devices must cooperate to offer meaningful services to users, regardless of technological choices such as service discovery and access protocols. Interoperability thus is a critical issue for the success of pervasive computing. In this context, we have previously introduced the MUSDAC middleware for interoperable service discovery and access across heterogeneous networks. Still, data exchanged in pervasive environments may be sensitive, and as service information is forwarded to unknown or untrusted networks, privacy issues arise. In this paper we present a privacy-aware solution for service discovery in heterogeneous networks, based on the MUSDAC platform. Specifically, we discuss privacy issues that arise during service discovery and mechanisms to control disclosure of private information contained in service-related data.

1 Introduction

Today, individuals carry various wireless-enabled multi-purpose digital devices, and applications that exploit their cooperative possibilities begin to emerge. Middleware systems are in particular being introduced to handle the richness of the services available in the environment, and the high heterogeneity of the various hardware, software and wireless network technologies. Heterogeneity has been primarily addressed by providing *multi-protocols* interoperability layers, and by managing network overlays atop dynamic *multi-networks* compositions. These solutions however further aggravate the issue of handling the multitude of networked services available to users.

A major trend to handle such service richness and provide localized scalability, is to rely on context information to infer mobile users' needs and autonomously locate the most appropriate services. However, as more applications and system services on mobile devices start to disseminate context data containing users' personal information (e.g., interests, location), it becomes critical to protect the users' privacy and therefore control this diffusion. One crucial middleware service in particular that must be considered is service discovery (SD), which provides information about the user's on-going interests and activities. Indeed, service discovery requests may disclose information such as user profiles, preferences, relations with other users, mobility pattern, and so on. Correlating multiple discovery requests from a single user would then quickly provide extensive knowledge about a user and reveal critical private information.

In this paper, we analyze how personal information is disclosed during service discovery in a heterogeneous environment, and propose various mechanisms to increase the level of privacy, and in particular prevent the correlation of multiple discovery requests. We base our work on the MUlti-protocol Service Discovery and ACcess (MUSDAC) middleware [13], a platform introduced to provide context-aware service discovery and access in pervasive environments by combining well-established patterns to address protocol interoperability (i.e., common representation) and multi-network discovery (i.e., ad hoc network composition). We therefore believe that the issues discussed and the mechanisms proposed are general enough and can be integrated also into other discovery platforms. In Sect. 2 we discuss challenges in service discovery for pervasive environments and analyze the MUSDAC platform impact on the privacy of clients and service providers. After that, we introduce in Sect. 3 the mechanisms that allow for privacy-aware multi-protocol service discovery. We assess our solution in Sect. 4 and present our concluding remarks in Sect. 5.

2 Service Discovery in Pervasive Environments

In this section, we first review the challenges and standard solutions for SD in multi-protocols, multi-networks environments, and present the MUSDAC platform. MUSDAC combines solutions for interoperability and multi-networking that specifically address the context requirement of pervasive computing. We then identify the inherent privacy issues caused by the use of such SD platforms.

2.1 The MUSDAC Platform

We assume an environment with various highly heterogeneous networks running the IP protocol, but without global IP routing. Each network is managed independently, and applications within each network may use various service discovery and service access protocols. We also assume that some devices with multiple network interfaces may connect to different networks simultaneously. In this context, the MUSDAC platform was designed to support the discovery of services (i) advertised using legacy SD protocols (interoperability) and (ii) hosted on any device in the environment (multi-networks).

A typical mechanism for supporting SD protocol interoperability is to rely on a common representation for service advertisements and discovery requests (either in the form of enriched advertisements and requests [1] or as sets of elementary events [2]). The interoperability layer may either be transparent for clients and services, with messages being translated on the fly between the different protocols [2], or may be explicitly accessed by clients through an API [7]. While the former approach generally leads to better performances and does not require any adaptation of the client applications, the latter allows the enhancement of service descriptions and requests with additional information (e.g., QoS, semantic). The dynamic composition of heterogeneous networks may be either achieved at the network level, providing a global IP network, or at the application layer with the deployment of a network overlay. Again, the network-level approach offers better performance and a straightforward support of legacy applications, but prevents each network from defining its own access policies and prevents the choice of a service instance based on communication cost and quality, or other contextual information related to the network path.

As previously established [3, 9], the use of context information is crucial in pervasive computing. We therefore designed MUSDAC to provide an explicit SD interface to clients, and to provide a SD overlay by managing the dynamic composition of nearby networks. MUSDAC is composed of (i) a set of Managers, each processing discovery and access requests of local and remote clients within a single network, (ii) sets of SD Plugins and Transformers, associated with each Manager, that interact with legacy SD protocols on behalf of their Manager, and (iii) Bridges that forward service discovery and access requests to remote Managers, enabling clients located in any network of the pervasive environment to use services available on another network. Figure 1 shows how these components interact. A more detailed presentation of the MUSDAC platform can be found in [13].

MUSDAC enables clients to add context information to their SD requests, and enables Bridges and Managers to exchange context information about the network characteristics and policies. This context information is used by Managers to select the most relevant services, and by Bridges to control the propagation of clients' requests. A complete description and evaluation of the context-awareness mechanism of MUSDAC is detailed in [14].

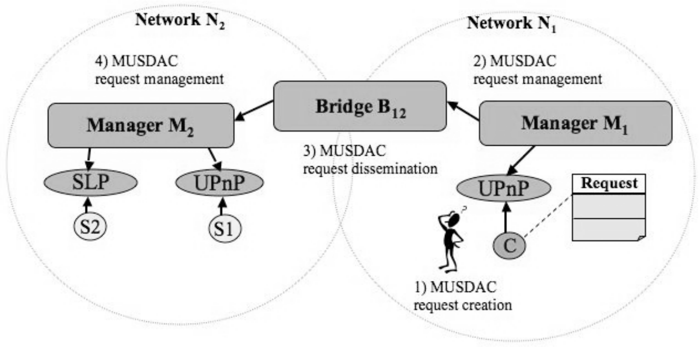
Network N2
Network N1
4) MUSDAC request management
2) MUSDAC request management
Bridge B12
Manager M2
Manager M1
3) MUSDAC request dissemination
SLP
UPnP
S2
S1
UPnP
Request
C
1) MUSDAC request creation

Fig. 1. Interaction Between MUSDAC Components

It is not required for each device in the environment to execute the MUSDAC platform. Indeed, a device may just host Manager/Plugins or the Bridge component. Within a network, one of the devices hosting a Manager component is elected to control the network, and registers the MUSDAC SD service to be used by clients using various SD protocols. This Manager in turn selects among the potential Bridges which one to activate based on various criteria (connectivity, performance, cost). While services just register with their preferred SD protocol, MUSDAC-aware clients first discover the MUSDAC SD service using any legacy SD protocol, and then use the MUSDAC API and formats to discover services in the pervasive environment.

2.2 A Trust Model for Multi-Network Service Discovery

Service discovery information is sensitive. Service request messages and service descriptions contain private information that can be later used to infer personal data such as users' desires and intentions or to link consecutive requests. Storage and analysis of service descriptions and requests over a period of time can increase the precision of inferred personal information. Service-related data hence must be carefully disclosed, particularly when it is going to be handled by untrusted entities. In the current MUSDAC middleware, users are assumed to trust all Bridges and Managers to fairly use their personal information. This trust model, however, is not appropriate for a real world multi-network scenario. Pervasive environments connect networks in different domains using distinct administration policies. Devices in remote networks may not have any reason to fairly use clients personal information or may even join the environment only to abuse personal data available through the platform.

We must define a trust model for MUSDAC consistent with the multi-network pervasive scenario. Consider a pervasive environment comprising n networks $N_1, ..., N_n$. Each network has a single Manager M_i responsible for handling requests from all clients belonging to network N_i. Two distinct networks N_i and N_j can be connected by a Bridge B_{ij}, in which case requests handled by M_i can be forwarded to M_j through B_{ij}. Our trust model is based on the following assumptions:

1. Clients in network N_i trust the local Manager M_i to fairly use their personal information. As clients and the Manager are located in the same network, their trust relationship can be created based either on a contract that defines how the Manager can use client information (and possible countermeasures in case of abuse), on the client's personal experience when using Manager M_i, or any other trust establishment protocol. In Fig. 1, client C trusts Manager M_1 to fairly use its personal information.
2. Clients in network N_i do not trust any Bridge $B_{kl}, \forall k, l \in \{1, ..., n\}, k \neq l$ to fairly use their personal information because they are not directly responsible for Bridge election and their requests are forwarded to Bridges regardless of their trust judgment. Besides that, clients do not know critical information about the networks connected by the Bridges such as topology or network technology in place, which are important to support the clients'

trust decisions. For instance, a client may not trust WiFi networks with low encryption levels. In Fig. 1, client C does not trust Bridge B_{12} to fairly use its personal information.

3. Clients in network N_i do not trust any other Manager M_j, $j \neq i$ to fairly use their personal information. As clients do not interact directly with remote Managers and their association is transient, it creates additional obstacles for trust establishment protocols (for instance protocols based on past experiences). In Fig. 1, client C does not trust Manager M_2 to fairly use its personal information.
4. All the entities in the environment trust that Bridges and Managers will correctly execute the MUSDAC protocols. We assume that every Manager can determine if Bridges are running the right software version and vice-versa. We also suppose that one MUSDAC component can detect another MUSDAC component misbehavior, and exclude ill-behaved components from the platform in later interactions. We are not investigating issues that arise when components maliciously run MUSDAC protocols.

2.3 MUSDAC Privacy Pitfalls

As the term privacy may assume different meanings depending on the research context, it is important to define its scope in the service discovery scenario. In this paper, privacy is defined as the *control over information disclosure*. As such, a privacy invasion is considered to occur when information regarding an entity is disclosed without the entity's explicit consent. When the trust model defined above is taken into account, it is possible to identify many channels for privacy invasions in the original MUSDAC design.

First of all, service requests are gradually flooded to MUSDAC components through successive multicasts: the local Manager forwards each client request to the group of Bridges connected to it, Bridges to neighbor Managers, and so forth until all the networks compatible with context rules and within a certain distance to the client are reached. As a result, personal information contained in requests is disseminated to multiple MUSDAC components even if relevant services have been found early in the discovery process.

Service descriptions may also contain private information about the service provider, and its access must be controlled. In pervasive computing, the roles of clients and service providers are not fixed, and users may even perform both at the same time. This creates yet another possibility for privacy invasions: service request data can be used along with service descriptions to identify a user and to infer more personal information, by combining both sources of private data. MUSDAC does not provide any specific mechanism to control access to service descriptions, so any service published by a user can be discovered by the whole environment and the service description can be potentially disclosed to a great number of entities, amplifying the privacy risks for the user.

Request routing can also be a vector for privacy invasion attacks. The current algorithm was designed to minimize MUSDAC components processing, by reducing the quantity of routing state information required to be cached in intermediary nodes. Addresses of Bridges and Managers along the path between a source and a destination Manager are appended to the request, so that they carry all information required for routing back the response. Routing data included in the request, however, can reveal details about the requester identity, such as the address of the source Manager and the path used to reach the destination Manager, and could be later used to relate consecutive service requests and infer clients' personal data.

Routing data is not the only source of identity information contained in service requests. Request contents may also contain personal data, especially if context information is included. For instance, requests from a client looking for content adapted to his mobile device can be linked to each other if the device model is not popular. As MUSDAC originally forwards requests in clear-text, all entities of a path have access to its contents and can use it to correlate distinct requests. Not every MUSDAC component needs to be able to read request details, though: Bridges, for instance, are only responsible for transport and should only access data relevant for routing.

3 Privacy-Awareness Mechanisms in MUSDAC

Privacy enhancements in MUSDAC aim at dissociating the source of a request from the request message and reducing the access of untrusted entities to service request contents. In this section, we present the modifications introduced into MUSDAC to achieve these goals. They can be divided into: (i) mechanisms to give clients control over their service discovery data propagation, presented in Sect. 3.1 and (ii) techniques to increase client privacy during request routing, introduced in Sect. 3.2.

3.1 Control over Discovery Data Propagation

MUSDAC's original design floods the networks with service requests and asynchronously waits for discovery results. Although this propagation strategy can produce results faster, it also exposes client's requests to a higher number of untrusted entities. According to the trust model defined in Sect. 2.2, clients trust only the local Manager to fairly use their personal information. The other entities, such as Bridges and remote Managers are not trusted by the client and can misuse request data to invade his privacy. The decision to trade-off performance for privacy should not be imposed by the platform, but rather left at the client's discretion. To enable clients and service providers to have more control over service discovery data disclosure, we enhance MUSDAC with mechanisms to define *how* requests must be propagated and *where* they should go.

Incremental Service Discovery: The original MUSDAC design provides only a **parallel** discovery strategy, which gradually floods platform components with client requests, reaching all significant networks. If sufficient results are found in networks closer to the client, the parallel strategy unnecessarily exposes personal information contained in service requests to all reachable MUSDAC components. To avoid needless service data disclosure, we propose two alternative propagation strategies: **progressive** and **one-by-one**. In the former, the local Manager initially sends the request to all the networks that are one hop away and waits for the results. If more results are necessary, the local Manager sends the request again to networks two hops away, and so forth. In the latter, the local Manager sends the request to a specific Manager one hop away and waits for results. If more results are needed, the request is sent to another Manager one hop away and so on, until all the Managers in neighbor networks are queried. Only then the request is sent to a Manager two hops away. Figure 2 shows the three strategies. The parallel strategy is asynchronous, so requests are forwarded independently of replies and quickly reach various MUSDAC components. The progressive strategy waits for results from networks one hop away before deciding if the discovery process should go on. Finally, the one-by-one strategy may disclose the request contents to a single Manager, if it provides sufficient results.

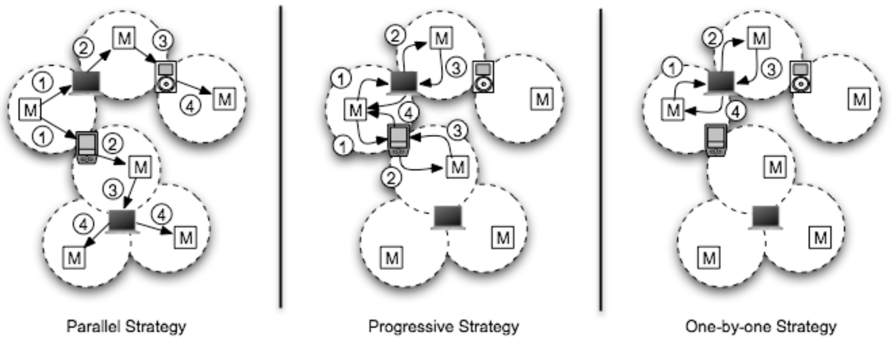

Fig. 2. Parallel, Progressive and One-by-One Propagation Strategies

Clients can define which strategy to use on a per-request basis. When using the progressive strategy, users can inform the number of results that must be obtained before stopping discovery and the maximum distance of Managers that should receive the request in terms of hops. With the one-by-one, strategy users can also define the maximum number of Managers that can receive the request. Implementation of new propagation strategies demands two modifications in MUSDAC. First, Bridges must be able to directly forward requests to other Bridges in order to bypass Managers that already processed the request. Second, a unicast mechanism is necessary to enable a Manager to send a discovery request to another specific Manager. Local Managers must have network topology information to be able to determine the route to a destination Manager. This is obtained by running a routing protocol based on OLSR (Optimized Link-State Routing) [8] on each MUSDAC component. Periodically, each component uses the beacon message to also advertise its neighbors. Other components use that data to compute a local view of the network and to determine how to route packets to specific destinations.

Privacy-Compliant Dissemination: The second mechanism introduced allows clients to control where their requests go. The original MUSDAC platform forwards requests to all connected networks, disregarding the clients' trust level on destinations. Nevertheless, users may establish trust relationships with companies or services, and expect those relations to be respected when using pervasive computing systems. This work extends the original MUSDAC platform by enabling service discovery to comply with two types of trust relationships concerning private data usage: between citizens and governments based on **privacy protection laws** and between consumers and corporations based on **privacy agreements**. Privacy protection law coverage is related to geographical boundaries, such as state, country, or group of countries. Privacy agreements on the other hand cover relations between a consumer and services provided by a company, possibly hosted in different domains. It is important to notice that, as service requests traverse several components before reaching the destination Manager, the trust relationship must hold for all the entities on the path, and not only between the client and the destination Manager.

To enable clients to define geographic and domain restrictions for service request propagation through the context-awareness mechanism of MUSDAC, Bridges must keep information about where they are located while Managers must store also the domains for which they are responsible. Service requests contain a list of geographical entities that adopted privacy laws trusted by the client and the list of companies that have privacy agreements accepted by the client. Privacy-related context is disseminated to neighbor MUSDAC components so that they can verify client-defined context rules to decide if the request can be forwarded or not. We developed two elementary ontologies for geographical and domain context data specification that increase the flexibility of this mechanism removing the need for exact matches between context rules and context data (recognizing that a request from a client who trusts the European Union privacy legislation can be forwarded to a Manager in France, for instance). Recent work in our group proposes directions on how to efficiently use ontologies for service discovery [10].

Privacy-related context data can also be used by service providers to control access to service descriptions. Service providers may want that only clients on a given geographical region under a certain legislation, or coming from a given domain administered by a trusted company are able to discover a service. In this case, the service provider includes in the service description its privacy preferences concerning domain and geographic location of the request source. Instead of requiring clients to include that information in the request – which would ultimately disclose their location and domain – service provider's privacy preferences are included in the discovery results. As they are forwarded back to the client, privacy preferences are checked and the response may be filtered out at one of the MUSDAC components along the path.

3.2 Privacy-Enhanced Routing

In Sect. 3 we stated that one of the goals of privacy-enhancements in MUSDAC is to dissociate the source of a service request from the request message. If a service needs to identify a client before allowing access, this identification should be explicitly performed at the application level, so that the client can evaluate what are the effects of revealing his identity, and act accordingly. Service discovery requests have two sources of identifiable information. First, request source address and routing information can reveal the request starting point. Second, service request contents along with source addresses can be used to correlate consecutive requests and reduce the set of possible origins of a message. We enhance MUSDAC with mechanisms to protect request source address and contents during multi-network message routing.

Hop-by-Hop Routing: Ideally, every communication should be anonymous, with identification provided by the application level. Anonymity, however, is a relative concept. An individual is never completely anonymous; his anonymity degree when performing some action depends on the anonymity set, the set of all possible subjects who could have caused the same action [11]. We are interested in reducing the probability of identifying the entity responsible for sending a service request by increasing the set of entities that could have sent the same request. However, it does not suffice to consider anonymity for each message independently. If different messages can be related, the anonymity set for the group of messages is smaller than the anonymity set of each message individually. Unlinkability [11] between messages is hence another important requirement to increase the anonymity set of multiple messages.

As described in Sect. 2.1, MUSDAC components add their identification to each message for routing purposes, including the local Manager that originated the request. Even though the client address is not included on service request messages, routing data narrows the set of possible sources and can be also used to correlate consecutive requests. We modify MUSDAC's original routing algorithm to increase the sender anonymity set for service discovery messages by performing hop-by-hop routing. After receiving a local request for a service, the local Manager creates a unique ID for the request and sends the requests to a neighbor Bridge. Before forwarding the request to the next MUSDAC component, the Bridge generates a new ID for the message and stores the original ID, the new ID and the local Manager address. This information is used afterwards to identify the reply message and send it back to the request source. Every other entity that forwards a message caches its original ID, new ID and source. This entry can be later deleted if no reply is received after a pre-defined timeout. As a result, intermediary components know only the previous and the next hops of a message. The sender anonymity set for the message increases on each hop, since all the messages coming from a MUSDAC component have the same source address information. The destination Manager, in particular, can only identify the last Bridge that forwarded the request.

End-to-End Anonymous Encryption: Even though some MUSDAC components are only responsible for routing packets during a service discovery, all of them have access to the request contents. Clients, however, may consider that service discovery information in a particular request is so sensitive that only the destination Manager should have access to the message contents. Still, clients must be able to receive discovery results without disclosing their identities. The straightforward solution of using public key encryption to protect service requests and discovery results, in that case, is unsuitable since it would require disclosure of client or local Manager identities. We add to MUSDAC two protocols for end-to-end service discovery encryption that provide clients with different anonymity levels.

A common solution to provide network communication anonymity is the use of mixes. They were first introduced by Chaum [5] as a mechanism to protect the relation between sender and receiver of a given message, providing *relationship anonymity* [11]. In its simplest form, it consists of a single trusted server that receives encrypted messages from different sources and forwards them to different destinations adding delays as needed, as illustrated by the left side of Fig. 3. To increase resistance against malicious mixes, Chaum proposed the use of cascade of mixes: users define a path of mixes and encrypt messages with different encryption layers, one with the public key of each mix on the path. As messages traverse mixes, encryption layers are removed and the message is delivered to the final destination, as showed by the right side of Fig. 3. This way, a single honest mix along the path suffices to guarantee the cascade secrecy.

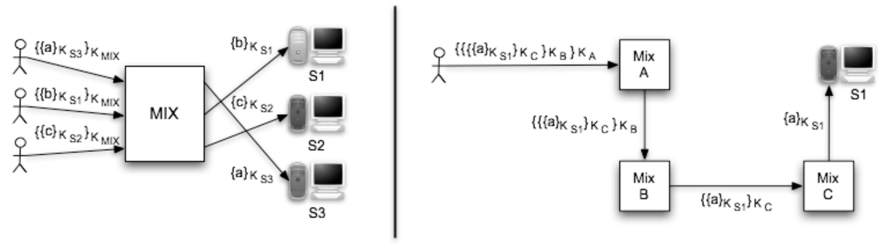

Fig. 3. A Simple Mix and a Cascade of Mixes

The privacy-aware version of MUSDAC uses the protocol proposed by Tor [6], a real world implementation of the Mix approach, to provide strong anonymity for service discovery. The protocol provides unilateral authentication (request sources remain anonymous), forward secrecy and key freshness. A local Manager increasingly establishes symmetric keys with every MUSDAC component on the path to a destination Manager, one at a time, using their public keys. Symmetric keys are used afterwards to encrypt messages with multiple layers, that are decrypted by a cascade of mixes. For instance, if Manager M_1 wants to establish a key with Bridge B_{12} that possesses public and private keys $\{K_{B_{12}}, K_{B_{12}}^{-1}\}$, it first sends a request containing the first half of a Diffie-Hellman key agreement protocol and a circuit ID, encrypted with the Bridge's public key ($M_1 \rightarrow B_{12} : \{g^{x_1}, ID_1\}_{K_{B_{12}}}$). The Bridge answers with the second part of the Diffie-Hellman key agreement in plain text, along with a hash of their common key k_1 ($B_{12} \rightarrow M_1 : g^{y_1}, H(k_1)$).

This protocol provides strong anonymity since only the destination manager is capable of accessing the request contents, and intermediary MUSDAC components cannot relate the source and destination Managers of the service request. If we take into account the environment dynamics, however, this protocol may be too costly for a single service discovery transaction. We also introduce a lighter version of the protocol that offers weaker anonymity but better performance for anonymous end-to-end encrypted service discovery. Instead of negotiating a key with every MUSDAC component on the path, a local Manager runs the above protocol only with the destination Manager and uses the agreed key to encrypt the service request. In this protocol, however, the relation between source and destination Managers of a request is no longer protected. The first MUSDAC component of the path, particularly, knows exactly the source and destination of a service request. Groups of compromised components can also reveal that information. Nevertheless, intermediary MUSDAC components are still unable to read the request neither the reply contents.

4 Solution Assessment

Privacy-protection usually has an effect over resource consumption. Some mechanisms proposed by this work, however, allow for a more rational use of network resources and may actually have positive impacts on the MUSDAC performance. In this section, we perform a qualitative and a quantitative evaluation of the privacy-enhancement mechanisms proposed. Whenever it is relevant, we discuss the impacts on users as well as on MUSDAC components.

4.1 Qualitative Assessment

Most part of the privacy protection mechanisms proposed in this work is not mandatory, and clients can use it independently or in combination to increase their control of how and when they want to release personal data. Implementation of these mechanisms requires the introduction of a pro-active routing protocol in MUSDAC. Nevertheless, routing data dissemination improves MUSDAC support for mobility. In the original design, routing information is appended to messages as they traverse the platform. If a modification happens to the route, such as a Bridge that moves to another location, the path stored in the message is not updated and the request or the response is lost. With routing dissemination in place, every MUSDAC component has a partial local view of the network and thus can determine the new location of another component that moved. When a message's next hop is no longer available, a MUSDAC component is able to find an alternative route to that hop.

Also, the progressive and one-by-one discovery strategies contribute to a better resource usage by reducing message processing during service discovery. Experimental data shows that inter-network connectivity degrades substantially after three Manager hops [12]. The two new dissemination strategies take that into account and provide a more rational use of resources by performing service discovery first in closer networks, and only forwarding service requests to other remote networks if more results are needed. As a consequence fewer service discovery messages are generated, offloading the platform components. Reduction of component resource usage is an important incentive for entities volunteering to run MUSDAC services. Even though those strategies take longer to produce discovery results when compared to the parallel strategy, this delay may be acceptable in service discovery since it is a non-interactive process. Furthermore, users needing faster responses can always choose to use the parallel strategy.

Malicious components can disrupt the protocol or abuse it to invade the privacy of clients. Misbehaving Bridges can ignore the propagation strategy chosen by the client and forward requests to every MUSDAC component on the network, or announce fake geographical or domain information to receive undue requests. To prevent these attacks, honest MUSDAC components can examine the behavior of neighbor components during protocol execution to identify if they are working as expected or not. For example, a Bridge that announces that is located on a given country but never returns results from that country may be advertising a false location, or two identical requests with different propagation strategies may suggest that one of them was modified by a malicious Bridge. An intrusion detection protocol could use this kind of information, provided by multiple MUSDAC components, to identify rogue protocol participants. After identification, the MUSDAC overlay network could be reconfigured to avoid malicious components or MUSDAC components could be adapted to tolerate misbehaving participants, for instance by using end-to-end encryption which is resistant even to groups of malicious nodes along a communication path.

4.2 Quantitative Assessment

Even though the greatest part of the privacy protection mechanisms in MUSDAC is optional, the new routing capabilities impose a permanent overhead on the architecture and can be considered as the only fixed cost of the new platform design in terms of resource utilization. We first analyze the quantitative impact of hop-by-hop routing on resource consumption and after we discuss the effects of introducing a pro-active routing protocol into MUSDAC. Finally, we detail the performance impact of the other remaining privacy-enhancing mechanisms.

Hop-by-hop routing requires MUSDAC components to cache routing data for each message they handle and to search cached data for next hop determination. This data consists of an original message ID, the message last hop identification, and the new message ID. Message IDs are 128-bit random values, while hop identifications are 128-bit MD5 hashes of Managers and Bridges network information such as domain name and IP address. Each MUSDAC component thus has to store 48 bytes of routing data per message and this data can be discarded after a pre-specified timeout. According to our experiments, a service discovery takes at most 2 seconds to finish [13]. Based on that data, we estimate that 30 seconds is an adequate timeout for routing data and we do not expect that MUSDAC components will have to process more than 50 messages during that time interval [12]. Bridges and Managers, hence, will have to store a route state table of at most 2.4 KB, which we do not consider as too constraining. Also, search and update operations on a table with 50 entries should not impose a great overhead on MUSDAC components, especially when taking into account that context-aware service request processing already takes at least 40 ms to complete on each MUSDAC component [14]. Regarding proactive routing, we use a protocol based on OLSR optimized to only discover routes at most four Manager hops long. Already existing beacon messages are used to disseminate route information causing only a small overhead on the beacon message size. We expect that Bridges and Managers on a pervasive environment will form a weakly connected graph. In that case, as MUSDAC components only store routes four hops long, the required storage space and processing to compute routing tables is not significant.

Besides this small but permanent overhead, other optional privacy features can affect service discovery performance. User-defined restrictions on service request propagation based on geographic and domain data is implemented as MUSDAC context rules. Previous results show that service discovery time increases by 1.0 ms to 1.6 ms for each context rule [14]. Based on that data, and as request propagation restrictions can be implemented by two context-rules, we expect that processing of service requests that define restrictions on propagation will be at most 3.2 ms slower, representing a processing time increase of 6.8% on each MUSDAC component per message, which we believe to be acceptable. As service providers may also specify privacy-related context rules, the total delay for processing a service request and its corresponding result is 6.4 ms for each hop and can be at most 51.2 ms for messages passing through 8 hops considering Bridges and Managers.

Service discovery strategies and end-to-end encryption introduce a higher delay for discovery results. The progressive strategy for service discovery can be much slower than the parallel strategy, especially for networks many hops away. However, for networks one hop away, their performance is identical, since all networks one hop away are discovered in parallel. The one-by-one strategy always performs poorer than the other two strategies, and the delay increases as the number of networks visited by the request also increases. Regarding MUSDAC components, the progressive and one-by-one strategies may require

Bridges and Managers to process the same message more than once, when discovering services in networks more than one hop away, but the protocol can be optimized to avoid this situation. Nevertheless, the user can choose among the three options, which one provides the best balance between privacy protection and performance according to the request privacy requirements.

Finally, end-to-end encrypted requests using the weaker anonymity protocol require local Managers to perform one public-key encryption and remote Managers to perform a public-key decryption operation per request. Public-key encryption using the RSA algorithm and 1024-bit keys can be achieved in less than 50 ms even in computers with processing power equivalent to today's mobile devices [15]. The processing cost of encrypting and decrypting service discovery requests with the agreed symmetric key is negligible. If the stronger anonymity protocol is used, two public-key operations are required for each message hop. As we expect service requests to traverse at most 8 hops (including Bridges and Managers), the total encryption overhead for a service request can be as high as 800 ms at the worst case. However, clients can limit end-to-end encrypted service discovery to neighbor networks or networks two-Managers away, and in that case the encryption cost would be of 400 ms at most per request. Table 1 summarizes the costs involved in adding privacy-protection mechanisms to MUSDAC. The table enumerates expected delays for the client and compares the overhead between the native and privacy-aware (P-A) versions of MUSDAC.

Table 1. Privacy Performance Overhead

	Client	MUSDAC	P-A MUSDAC
Hop-by-hop Routing (storage)	-	0 KB	3.2 KB
Privacy Related Context	6.4 ms - 51.2 ms (per request)	2.1% - 3.4% (per rule)	6.8% (plus rules)
Progressive Service Discovery	0 - 4 times slower	-	-
One-by-one Service Discovery	0 - n times slower (for n networks)	-	-
End-to-end encryption (weaker)	100 ms (per request)	0 ms (no encryption)	50 ms (Mgrs.) 0 ms (Bridges)
End-to-end encryption (stronger)	200 ms - 800 ms (per request)	0 ms (no encryption)	50 ms

5 Conclusion

Interoperability between heterogeneous devices, networks and protocols is a fundamental factor for the success of pervasive environments. As we move towards this objective and the flow of information among different systems and applications is simplified, privacy issues arise. In service-oriented architectures, particularly, data associated to services such as service descriptions and requests become accessible to a greater number of entities. This data can contain sensitive information and malicious entities may abuse it to infer personal details such as activities and preferences.

In this work, we discussed privacy issues raised by service discovery in pervasive environments, particularly when context information is used to increase the relevance of the results. To address these issues, we proposed a trust model consistent with multi-protocol and multi-network service discovery, identified four complementary mechanisms to increase the privacy protection of mobile users interacting with the environment and performed an initial assessment to evaluate the impact of implementing those features. Figure 4 summarizes the mechanisms we propose (solid lines) and the modifications necessary to implement them on MUSDAC (dashed lines).

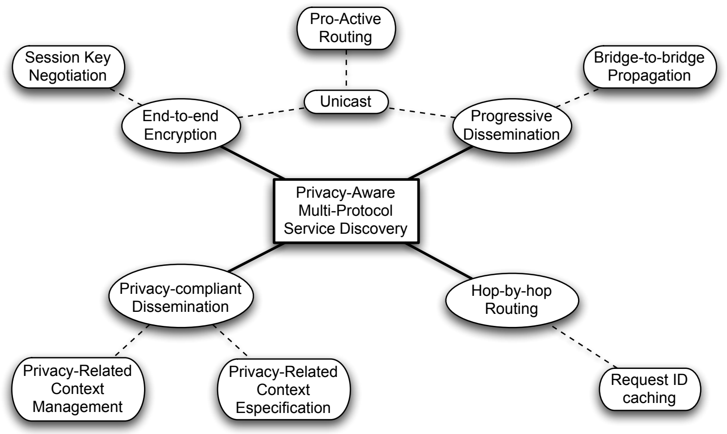
Pro-Active Routing
Session Key Negotiation
Bridge-to-bridge Propagation
Unicast
End-to-end Encryption
Progressive Dissemination
Privacy-Aware Multi-Protocol Service Discovery
Privacy-compliant Dissemination
Hop-by-hop Routing
Privacy-Related Context Management
Privacy-Related Context Especification
Request ID caching

Fig. 4. Privacy-Awareness Features and Required MUSDAC Modifications

Even though our implementation is based on a specific platform, we believe that the problems discussed in this paper are common not only to other middleware for multi-protocol service discovery but also to other systems that distribute data to heterogeneous networks, such as content-based networking [4]. We also believe that the solutions proposed can be easily adapted to increase privacy protection on such systems. As part of the IST PLASTIC project[1] we are developing a privacy-aware service discovery service based on the MUSDAC platform, and also studying how to enhance other systems that present similar characteristics with privacy-awareness features.

Acknowledgment This work is part of the IST PLASTIC project and has been funded by the European Commission, FP6 contract number 026955.

[1] `http://www.ist-plastic.org/`

References

1. J. Allard, V. Chinta, S. Gundala, and G. G. Richard III. Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability. In *SAINT '03: Proceedings of the 2003 Symposium on Applications and the Internet*, January 2003.
2. Y.-D. Bromberg and V. Issarny. INDISS: Interoperable Discovery System for Networked Services. In *Proceedings of the 6th International Middleware Conference*, November 2005.
3. L. Capra, S. Zachariadis, and C. Mascolo. Q-CAD: QoS and Context Aware Discovery Framework for Mobile Systems. In *Proceedings of the International Conference on Pervasive Services (ICPS'05)*, July 2005.
4. A. Carzaniga and A. L. Wolf. Content-Based Networking: A New Communication Infrastructure. In *IMWS '01: Revised Papers from the NSF Workshop on Developing an Infrastructure for Mobile and Wireless Systems*, October 2001.
5. D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2), February 1981.
6. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX*, August 2004.
7. A. Friday, N. Davies, N. Wallbank, E. Catterall, and S. Pink. Supporting Service Discovery, Querying and Interaction in Ubiquitous Computing Environments. *ACM Baltzer Wireless Networks (WINET) Special Issue on Pervasive Computing and Communications*, 10(6), November 2004.
8. P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Proceedings of IEEE INMIC 2001*, December 2001.
9. C. Lee and S. Helal. Context Attributes: An Approach to Enable Context-awareness for Service Discovery. In *SAINT '03: Proceedings of the 2003 Symposium on Applications and the Internet*, January 2003.
10. S. Ben Mokhtar, A. Kaul, N. Georgantas, and V. Issarny. Efficient Semantic Service Discovery in Pervasive Computing Environments. In *Proceedings of the 7th International Middleware Conference (Middleware'06)*, December 2006.
11. A. Pfitzmann and M. Köhntopp. Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In *International Workshop on Designing Privacy Enhancing Technologies*, July 2000.
12. P.-G. Raverdy, S. Armand, and V. Issarny. Scalability Study of the MUSDAC Platform for Service Discovery in B3G Networks. In *Proceedings of Wireless World Research Forum Meeting (WWRF-17)*, November 2006.
13. P.-G. Raverdy, V. Issarny, R. Chibout, and A. de La Chapelle. A Multi-Protocol Approach to Service Discovery and Access in Pervasive Environments. In *Proceedings of MOBIQUITOUS - The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services*, July 2006.
14. P.-G. Raverdy, O. Riva, A. de La Chapelle, R. Chibout, and V. Issarny. Efficient Context-Aware Service Discovery in Multi-Protocol Pervasive Environments. In *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, May 2006.
15. M. J. Wiener. Performance Comparison of Public-Key Cryptosystems. *RSA Laboratories' CryptoBytes*, 4(1), July 1998.