Proposal for Improving Google A2A Protocol: Safeguarding Sensitive Data in Multi-Agent Systems

Yedidel Louck[1,2], Ariel Stulman[2], and Amit Dvir[1]

[1]Department of Computer and Software Engineering, Ariel Cyber Innovation Center, Ariel University, Israel
[2]Department of Computer Science, Jerusalem College of Technology, Israel

June 18, 2025

Abstract

A2A, a protocol for AI agent communication, offers a robust foundation for secure AI agent communication. However, it has several critical issues in handling sensitive data, such as payment details, identification documents, and personal information. This paper reviews the existing protocol, identifies its limitations, and proposes specific enhancements to improve security, privacy, and trust. It includes a concrete example to illustrate the problem and solution, research-backed rationales, and implementation considerations, drawing on prior studies to strengthen the arguments and proposed solutions. This proposal includes seven enhancements: short-lived tokens, customer authentication (SCA), granular scopes, explicit consent, direct data transfer, multi-transaction approval, and payment standard compliance. The vacation booking example illustrates how these enhancements reduce risks and enhance user experience.

1 Introduction

The rapid appearance of *Agentic AI* autonomous systems that plan, delegate, and collaborate without human intervention has created an urgent need for robust, standardized communication protocols [14]. Google's A2A (Agent-to-Agent) protocol [48] establishes such a foundation by defining a declarative, identity aware framework for discovering, authenticating, and exchanging tasks among heterogeneous agents. A2A's core mechanism as shown at Figure 1, the `AgentCard`, provides machine readable metadata that enables seamless interoperability across organizational and technical boundaries. Complementing A2A, the Model Context Protocol (MCP) standardizes the integration of large language models (LLMs) with external data sources and tools, allowing agents to leverage these models for real-time contextual understanding during task execution. This integration enhances agents' capabilities in complex scenarios, supporting use cases ranging from payment orchestration to business automation [26]

However, as multi-agent ecosystems scale in complexity, they also open new attack surfaces. Recent threat analyses of both A2A and the related MCP reveal sophisticated

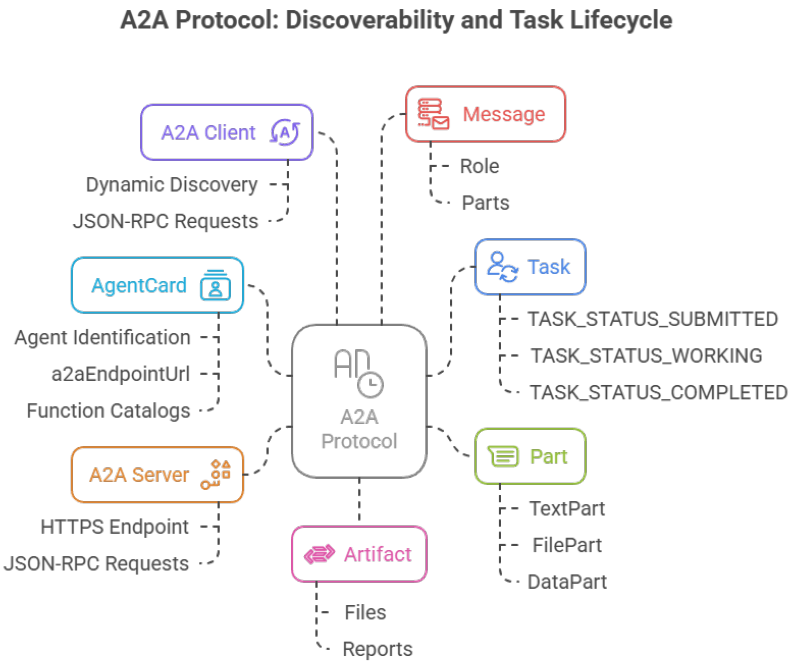

Figure 1: A2A's core mechanism

vectors such as *shadowing* (where malicious tool descriptions subvert genuine workflows), *tool poisoning*, and *naming attacks* that exploit implicit trust in discovery mechanisms [29, 44]. Furthermore, broad surveys of AI agents underscore systemic risks-ranging from prompt injection to cross-agent privilege escalation-that can lead to data exfiltration or unauthorized task execution [20, 32, 45]. Similarly, "enterprise-grade" analyses of MCP implementations demonstrate that without rigorous, zero-trust controls, attackers may exploit unsecured tool endpoints or manipulate context payloads to bypass security checks [37].

Taken together, these findings highlight a gap between protocol design and real-world resilience: A2A and MCP provide the plumbing for agent interoperability, but they do not yet enforce the fine-grained confidentiality, integrity, and consent guarantees required for sensitive data exchange. In this paper we review six critical weaknesses in the handling of personal information, such as payment credentials, user identity and sensitive documents, are identified. Building on proactive threat models and industry best practices, we propose seven concrete enhancements ranging from short-lived, client bound tokens to explicit user consent fields and direct data transfer processes to harden A2A communications against the most urgent threats of the day. This proposal includes a concrete example to illustrate the problem and solution, research-backed rationales, and implementation considerations, drawing on previous studies to strengthen the arguments and proposed solutions.

2 Overview of the A2A Protocol

Google's A2A (Agent-to-Agent) protocol [48] builds upon widely adopted web standards HTTP, HTTPS, JSON-RPC, and Server-Sent Events (SSE) to provide an extensible,

interoperable framework for autonomous agent communication. Agents establish mutual authentication via OAuth 2.0 [27] flows and JSON Web Tokens (JWTs) [30], while RSA key pairs are used for signature validation and secure key exchange. Specifically, OAuth 2.0, an industry-standard authorization framework, enables agents to securely delegate access to each other's resources without sharing credentials, using standardized flows like the authorization code grant. Meanwhile, JWTs, compact and cryptographically signed tokens, facilitate the secure transmission of authentication claims between agents, ensuring both integrity and verification of access rights during interactions. Role-based access control (RBAC) is natively supported through user- and agent-scoped JWT claims, and fine-grained permissions map directly onto task and message schemas to enforce the principle of least privilege. By default, A2A messages carry only minimal metadata (e.g., action identifiers, input/output schemas, and consent fields), reducing sensitive data exposure when agents negotiate capabilities or invoke third-party services such as payment gateways or AI inference backends.

A2A's flexibility underpins diverse real-world scenarios:

- **Service Booking:** Agents coordinate multi-step operations (e.g., flight, hotel, transportation) by passing structured requests containing specific payment tokens and availability parameters.
- **Enterprise Task Management:** Supply-chain or project-management agents exchange only the data needed to complete discrete tasks, minimizing shared context.
- **Cross-Provider Collaboration:** Agents from different platforms (e.g., Google, PayPal, Cohere) interoperate securely via standardized `AgentCard` discovery and encrypted channels.

Despite these strengths, A2A inherits an inherent trade-off between security and efficiency: tightening authorization or adding cryptographic checks invariably increases latency and complexity, whereas relaxing controls risks unauthorized data flows [42]. Crucially, A2A currently lacks specialized safeguards for handling particularly sensitive payloads-such as payment credentials or identity documents-beyond generic token expiry. A similar concern is raised in "AI Agents Meet Blockchain" observes, decentralized agent ecosystems without centralized guardrails require robust consent, auditing, and privacy-preserving delegation mechanisms to maintain trust [32]. Without such controls, adversarial agents might exploit overly broad scopes or unmonitored message channels to exfiltrate or tamper with confidential information. The following sections will analyze these gaps in more detail and propose targeted enhancements to ensure that A2A can securely mediate sensitive data exchanges at scale.

3 Identification of Issues in Handling Sensitive Data

Despite its benefits, A2A has some critical problems in the handling of sensitive data such as payment details, ID documents and personal data. The following six issues are based on real world experience, peer reviewed literature and documented CVE threats.

3.1 Absence of limitations on token lifetime

Although A2A is based on OAuth 2.0, it does not enforce strict expiration durations (e.g. seconds or minutes) for tokens used in sensitive transactions. Without such restrictions,

leaked tokens may remain valid for hours or even days, increasing the risk of unauthorized reuse. revocation. As demonstrated in "AgNet" [25], long-lived tokens are a systemic weakness in distributed architectures, allowing for multiple accesses in the event of a compromise. For example, CVE-2025-1198 shows that revoked GitLab personal access tokens are still accepted by long-term ActiveConnection [11]. Another case described by CVE-2025-1801, where a low-privileged user obtained a JWT issued to a high-privileged user on account [12].

3.2 Lack of Strong Customer Authentication (SCA)

The A2A protocol does not have built-in requirements for strong authentication, such as two-factor or biometric authentication, for high-value transactions such as payments or identity switching. Without these safeguards, adversaries may perform unauthorized acts on behalf of the user [24]. The Medibank breach of 2022, where attackers gained access to personal data of 9.7 million people through the lack of multifactor authentication, illustrates the tangible consequences of [28]. CWE-306 [10] also classifies systems that do not authenticate users before performing critical functions as inherently vulnerable. In addition, the AI Agents Meet the Blockchain project (aclm) [32] proposes Zero Knowledge Proof (ZKP) techniques for secure authentication in decentralized environments, but the A2A does not include such a mechanism.

3.3 Insufficiently Granular Token Scopes

Tokens in A2A do not define precise ranges for sensitive transactions, which introduces the risk of privilege escalation. For example, a token issued to initiate a payment may inadvertently grant access to unrelated data. The study on the multi-agent security tax [42] argues that coarse-grained authorization models increase the probability of data exposure. This concern is highlighted by vulnerability CVE-2023-4456 [4] on LokiStack. Similarly, CWE-1220 [8] documents the lack of granularity in access control policies leading to infringements of the principle of the minimum privilege.

3.4 Lack of Transparency and User Consent

A2A lacks mechanisms that notify users about, or request consent for, the sharing of sensitive data with agents. "AI Agents Meet Blockchain" [32] emphasizes the importance of user transparency in decentralized agent-based environments. CVE-2024-44131 illustrates the consequences of bypassing such protections: malicious applications were able to circumvent Apple's Transparency, Consent, and Control (TCC) framework to access sensitive data without user approval [5]. CWE-200 [9] codifies this as unauthorized exposure of sensitive information.

3.5 Potential Excessive Exposure of Data to Agents

Agents in A2A ecosystems can access significantly more data than is necessary. "AI Agents Under Threat" [20] articulates how agent2agent data propagation can lead to unintended sharing of sensitive information. CVE-2023-41745 [3] and CVE-2022-45449 [2] document such exposures due to excessive privilege allocation or excessive data collection.

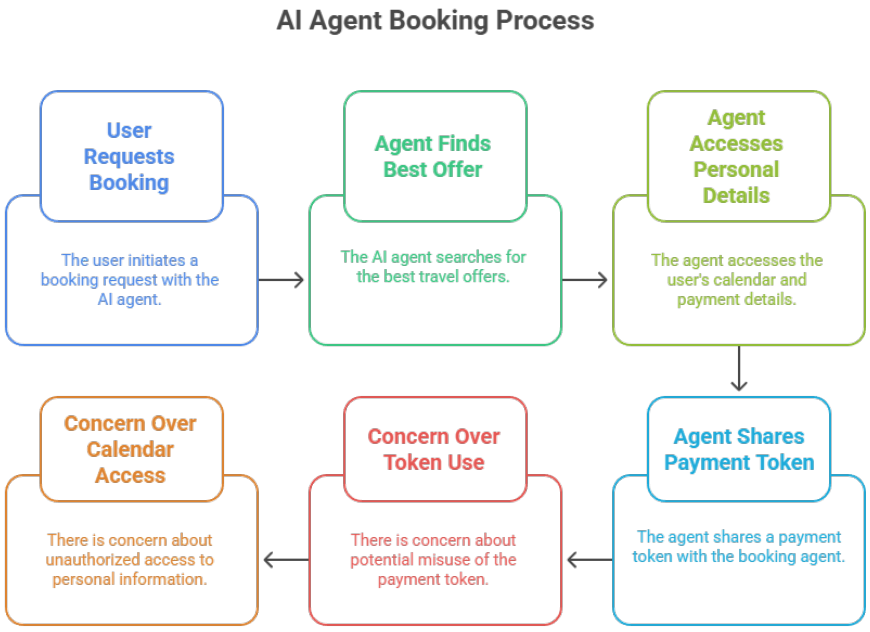

Figure 2: Vacation Booking Process Using An AI Agent

3.6 Risk of Data Disclosure to the Agent Itself

Moreover, the problem is not only that information is being disclosed to malicious actors, but that the very act of disclosing information to the agent itself is problematic, and that even well-meaning agents cannot be presumed to be safe from interference. Prompt injection attacks illustrate how malicious inputs can harvest classified information. "AI Agents Under Threat" [20] warns of this threat. CVE-2024-7042 [7] and CVE-2024-45989 [6] demonstrate how AI agents were exploited to perform unauthorized actions and exfiltrate sensitive data.

3.7 Concrete Example: Vacation Booking

Below is a usage scenario for the protocol without the proposed improvements. As reflected in Figure 2

- The user requests an AI agent to book a vacation, including flights, hotel, and taxi, using his calendar, personal information and payment details.
- The agent shares a payment token with the booking agent, valid for hours or days.
- The agent accesses the user's full calendar, including irrelevant personal details (e.g., medical appointments), and payment details.

Concern: The token could be used for additional payments, accidentally or intentionally, and calendar access exposes private information without explicit consent.

4 Proposed Enhancements to the Protocol

To address the identified issues, we propose seven enhancements to the A2A protocol, integrating research-backed solutions and aligning with advanced standards:

1. **Short-Lived Tokens for Sensitive Operations**
   Description: Mandate that tokens for sensitive operations, such as payments or ID transfers, have very short lifetimes (30 seconds to 5 minutes). Tokens expire after a single use or predefined time, requiring re-authentication for further actions.
   Rationale: As demonstrated in "AgNet" [25], short-lived tokens reduce the risk of unauthorized or repeated access. In our case, a payment token expiring quickly limits the window for leaks. "AI Agents Meet Blockchain" [32] supports this by noting that encryption and temporary records in blockchain systems enhance privacy, and our proposed tokens align with this principle.

2. **Strong Customer Authentication (SCA) for Sensitive Transactions**
   Description: Implement a mechanism requiring SCA, such as SMS codes, biometric verification, or bank login, for every sensitive transaction. This can be integrated into the task flow, ensuring user authentication before actions.
   Rationale: "AI Agents Meet Blockchain" [32] highlights that ZKPs enable secure authentication without exposing data, and in our case, SCA based on ZKPs or similar techniques ensures only verified users authorize sensitive actions. This aligns with PSD2 requirements for payment transactions, ensuring regulatory compliance.

3. **More Granular Token Scopes**
   Description: Extend OAuth 2.0 scopes to define precise permissions, such as "payment of $1000 to Hotel X on Date Y" or "access to calendar availability only." This prevents token use for unauthorized actions.
   Rationale: "Multi-Agent Security Tax" [42] notes that high granularity in permissions reduces unnecessary data exposure. In our case, precise scopes ensure agents access only required data, aligning with GDPR's Minimum Necessary Disclosure principle.

4. **Explicit User Consent Mechanism**
   Description: Add a "consent" field to message or task structures, requiring agents to obtain explicit user approval before sharing sensitive data, specifying data type, purpose, and recipient.
   Rationale: "AI Agents Meet Blockchain" [32] emphasizes transparency as critical for trust in decentralized systems. In our case, explicit consent enhances user trust and ensures GDPR compliance, reducing the risk of unapproved data sharing.

5. **Direct Data Transfer**
   Description: Enable direct transfer of sensitive data between the user and service provider, bypassing intermediary agents. For example, in payments, the agent redirects the user to the bank for direct transfer, with user verification.
   Rationale: Inspired by "Multi-Agent Security Tax" [42]'s "active vaccines" to prevent malicious data spread, direct transfers avoid exposing data to intermediaries. "AI Agents Under Threat" [20] supports this by emphasizing the need to limit data propagation, and in our case, direct transfers prevent the exposure of sensitive data to the agent and reduce the risks of leakage.

6. **Support for Multi-Transaction Approval**
   Description: Allow a single user approval for a series of related transactions (e.g. flight, hotel, taxi payments), with tokens restricted to that series, short-lived, and SCA-verified.
   Rationale: This balances security and convenience, as recommended in "Multi-Agent Security Tax" [42] for efficiency-security trade-offs. In our case, it reduces repeated authentications while maintaining strict token restrictions. Additionally, this solution addresses the well-documented issue of "consent fatigue," where users, overwhelmed by frequent approval requests, may approve actions without proper scrutiny, increasing security risks. As highlighted in the Palo Alto Networks article [53] repeated requests can lead to user desensitization, potentially resulting in the approval of critical actions like write operations without adequate attention, a phenomenon similar to MFA fatigue. By consolidating approvals for related transactions, our solution minimizes the frequency of requests, thereby reducing the risk of exploitation by malicious actors while enhancing the user experience.

7. **Compliance with Payment Standards**
   Description: Ensure the protocol supports standards like PSD2, requiring SCA for certain transactions, by defining interfaces for payment agents that incorporate these requirements.
   Rationale: Compliance with regulations is critical for adoption, as noted in "AI Agents Meet Blockchain" [32]. Our proposal ensures A2A aligns with regulatory requirements, enhancing its reliability.

4.1 Application to the Example

Protocol Usage Scenario After The Proposed Improvements:

1. The user requests the AI agent to book a vacation, including flights, hotel, and taxi.
2. The agent requests permission to access calendar availability only, displaying: "I will share availability dates with the booking agent. Approve?" The user approves.
3. The agent finds a suitable booking and requests payment approval: "I will process $1000 for Hotel X, $500 for Flight Y, and $50 for Taxi Z. Approve?" The user approves.
4. The agent requests a 5-minute token from the user's bank, scoped to "payment for vacation X on Date Y."
5. The bank issues a 5-minute token for the approved transaction series.
6. The agent shares the token with the booking agent.
7. The booking agent requests the bank to process payments using the token.
8. The bank prompts the user to verify the action via SMS code or biometric authentication.
9. The user approves, and payments are processed.
10. The token expires, preventing reuse.

Advantages:

- **Short-Lived Token**: Prevents reuse or unauthorized access.
- **SCA**: Ensures user approval for each payment.
- **Precise Scope**: Token restricted to approved transactions.
- **Explicit Consent**: The user is informed and approves the sharing of data.
- **Minimal Access**: The agent accesses only availability, not the full calendar.
- **Direct Transfer**: In order to protect sensitive information from agents, payment is sent straight from the bank to the booking agent.

4.2 Research Base Rationale for This Proposal

This proposal is based on previous research and enhances A2A's security, privacy, and efficiency:

- **Enhanced Security**: Short-lived tokens and SCA reduce leak risks, as recommended in [20, 32, 25].
- **Regulatory Compliance**: SCA and PSD2 support ensure compliance, as emphasized in [32].
- **Transparency and Trust**: Explicit consent increases user trust, per [32].
- **Flexibility and Efficiency**: Multi-transaction approval balances security and convenience, as suggested in [42].
- **Common Standarts**: The enhancements integrate with OAuth 2.0 and PSD2, as proposed in [25].
- **Exposure Prevention**: Direct transfers minimize leak risks, according to [20, 42].

5 Evidence-Based Support for Proposed Enhancements

In order to confirm the technical rationale behind the proposed improvements to the Google A2A Protocol, this section provides a literature review of each of them. Each section focuses on one proposed improvement and summarises previous literature reviews showing that the improvements are not only technically motivated but also based on established research in areas such as authentication protocols, cryptography, distributed identity systems and access control. Together, these references provide empirical and architectural support for the adoption of the proposed measures.

5.1 Use of Short-Lived Access Tokens

One of the core enhancements we proposed to improve the security of Google's A2A protocol is the mandatory use of short-lived access tokens. This design decision substantially reduces the window of opportunity for replay attacks and limits the utility of any compromised credential. Multiple studies affirm this principle. Teng (2023) introduced *ActionID*, a machine-to-machine authentication protocol that issues tokens tightly scoped in both action and time. The author notes that "the short-lived token limits the time window for action execution," thereby reducing the effectiveness of credential theft [49]. Similarly, Ohwo et al. (2024) proposed a blockchain-based smart home access system in which "short-lived tokens [are] used to mitigate risks like replay attacks and user profiling" [40]. These tokens expire rapidly, neutralizing any intercepted credentials. Narajala et al. (2025) extended this logic to GenAI multi-agent systems, proposing a just-in-time registry that dynamically provisions tokens only for the duration of a tool invocation. This architecture, they argue, "minimizes the attack surface associated with persistent credentials by dynamically provisioning short-lived access tokens only when needed" [38]. Finally, Xiao et al. (2024) addressed the same vulnerability in the IoT domain, where resource-constrained devices are especially susceptible to token compromise. They developed *MCU-Token*, which binds a short-lived, hardware-derived token to each request, thereby making every token instance single-use and resistant to replay [51]. Together, these findings offer both conceptual and empirical support for implementing short-lived tokens in the A2A protocol as a fundamental safeguard against credential replay and session hijacking.

5.2 Strong Customer Authentication (SCA) for Sensitive Transactions

Implementing Strong Customer Authentication (SCA) for sensitive operations in the A2A protocol, such as payments and identity verification, significantly improves protection against impersonation and unauthorized transactions. Recent research underscores the feasibility and necessity of using modern, privacy-preserving SCA mechanisms such as zero-knowledge proofs (ZKPs), biometrics, and multi-factor authentication (MFA). For example, Neera et al. propose a mobile payment protocol that leverages ZKPs and identity-based signatures to verify user identity without revealing sensitive data while ensuring compliance with financial regulations such as PSD2 [39]. Their scheme guarantees cryptographically enforced SCA while maintaining user privacy, demonstrating that SCA can be both secure and regulatory compliant. Ahmad et al. present BAuth-ZKP, a blockchain-based MFA framework using smart contracts and ZKPs to authenticate users in smart city environments [15]. This design proves the viability of decentralized, privacy-respecting SCA mechanisms, directly applicable to agent-mediated A2A architectures. On the biometric front, Gernot and Rosenberger introduce a technique for generating one-time biometric templates, mitigating the risk of replay attacks by ensuring biometric credentials are valid only once [22]. This reinforces the 'inherence' factor of SCA in a technically sound manner. Finally, Lyastani et al. empirically demonstrate that while 2FA dramatically improves account security, inconsistent or poorly designed SCA flows reduce adoption [23]. Thus, their findings support designing usable, transparent SCA processes for sensitive actions in A2A. Collectively, these studies provide strong empirical and architectural justification for integrating SCA into the A2A protocol, en-

suring that sensitive transactions are approved only by verified users under secure and user-friendly conditions.

5.3 More Granular Token Scopes

Enforcing fine-grained token scopes in the A2A protocol is essential for upholding the principle of least privilege and minimizing the exposure of sensitive resources during agent-to-agent transactions. The risks of overly broad permissions in bearer tokens are well-documented in recent research. Cao et al. propose a stateful, least-privilege authorization model that allows client-side applications to dynamically constrain the scope of OAuth tokens using WebAssembly-based privilege attenuation policies [18]. Their system empowers developers to explicitly encode minimal access rights per session, enabling secure delegation without overprovisioning. This aligns directly with the A2A context, where agents must act within strict permission boundaries to avoid inadvertent data access. Complementarily, South et al. extend OAuth 2.0 and OpenID Connect for authenticated agent delegation, introducing agent-specific credentials with precise, auditable scope limitations [47]. Their framework demonstrates how user intent can be translated into tightly scoped permissions, ensuring that AI agents operate only within authorized domains. These works collectively support the need for scope-constrained access tokens as a foundational safeguard in agent-mediated architectures. Dimova et al. further substantiate this by showing that 18.5% of OAuth deployments on the web request unnecessary scopes, violating the GDPR's minimum necessary data principle [21]. Kaltenböck et al. reinforce this position by embedding scope-aware policies within a Zero Trust single sign-on framework, emphasizing explicit scope definitions to limit token capabilities at authentication time [31]. Altogether, the literature affirms that fine-grained token scopes are not only technically viable but also indispensable for ensuring data minimization, secure delegation, and regulatory compliance in distributed, AI-driven authentication systems.

5.4 Explicit User Consent Mechanism

Embedding an explicit user consent mechanism within the A2A protocol is critical for aligning with privacy regulations such as GDPR and for building user trust in agent-mediated systems. Recent research emphasizes that consent must be freely given, informed, specific, and revocable throughout the data lifecycle. Merlec et al. propose a blockchain-based dynamic consent management system, where users can grant, audit, or withdraw their consent via smart contracts stored on a tamper-proof ledger [36]. This architecture guarantees accountability, traceability, and user autonomy in data-sharing environments. Complementarily, Khalid et al. formalize the security and privacy requirements of such systems, proposing the integration of zero-knowledge proofs and cryptographic primitives to ensure that consent is provable, minimal, and compliant by design [33]. Their work outlines how systems can enforce consent boundaries while maintaining confidentiality. In a multi-agent context, Xu et al. demonstrate that autonomous privacy agents can enforce user-defined consent policies, making decisions that reflect GDPR principles such as data minimization and informed processing [52]. These agents act only within verified constraints, providing technical assurance that consent decisions are respected. Finally, Pathmabandu et al. introduce a consent management engine that enables granular, real-time visibility into data collection within IoT systems [41]. Their

engine offers digital nudging and fine-grained control, reinforcing the user's right to control their personal data. Collectively, these studies support the integration of explicit, technically enforceable consent mechanisms into A2A-style protocols, ensuring that sensitive data is shared only with informed user approval, and that such actions remain transparent and auditable.

5.5 Direct Data Transfer

In the evolving landscape of multi-agent systems, ensuring the security of sensitive data during interactions between users and service providers is paramount. Traditional multi-agent architectures often involve intermediaries that can pose significant security risks, such as data leakage or unauthorized access. To address these concerns, the proposed enhancement advocates for direct data transfer, where sensitive information is exchanged directly between the user and the service provider, bypassing intermediary agents. This approach is supported by recent research in multi-agent security. Firstly, [42] introduces the concept of "active vaccines" to prevent the spread of malicious prompts in multi-agent systems, underscoring the need to minimize intermediary involvement to reduce systemic vulnerabilities. Similarly, [20] highlights the unpredictability of multi-step user inputs and the complexity of internal executions in AI agents, emphasizing the necessity of limiting data propagation to trusted entities. Furthermore, [19] discusses the open challenges in securing systems of interacting AI agents, particularly the threats arising from free-form interactions and network effects that can amplify security breaches, which direct data transfer mitigates by reducing intermediaries. Lastly, [47] presents a framework for authenticated delegation and authorized AI agents, which can be extended to support direct data transfer by ensuring secure, intermediary-free communication channels. On the basis of these studies, it can be concluded that direct transmission of data increases the security of multi-agent systems by reducing reliance on intermediaries and thus minimising the risks of data exposure and unauthorised access. This improvement is essential for applications involving sensitive data such as payments and personal data, where security and privacy are paramount.

5.6 Support for Multi-Transaction Approval

In the area of multi-agent systems and financial technology, securing financial transactions while preserving user comfort is a key challenge, especially for sensitive data such as payment details. The proposed enhancement, 'Multi-Transaction Approval,állows for a single authorization for a series of transactions using limited-time tokens validated by strong customer authentication. This approach balances security and usability by reducing the need for multiple authentication while maintaining strict security measures, thus increasing the effectiveness of multi-step workflow in agent-based systems. Recent research on the security of financial technology supports this improvement. For instance, [34] underscore the pivotal role of multi-factor authentication (MFA) in securing mobile financial transactions, advocating its use to mitigate fraud risks, which aligns with the SCA requirements of the proposed enhancement. Similarly, [50] provide a systematic review of MFA in digital payment systems, noting that grouping transactions under a single secure authentication session is feasible with robust mechanisms, supporting the enhancement's design. Furthermore, [13] propose a framework integrating MFA with machine learning to secure online financial transactions, adaptable to multi-transaction

approval by ensuring SCA verification for each transaction in a series and employing anomaly detection to enhance security. Additionally, [16] develop an MFA algorithm for mobile money applications, combining multiple authentication factors to secure transactions, which can be extended to support a single approval for multiple transactions as proposed. In conclusion, the "Multi-Transaction Approval" enhancement is robustly supported by current research in financial technology security, emphasizing advanced MFA techniques and fraud detection methods. By implementing this enhancement, the A2A protocol can offer a secure and user-friendly approach to managing multiple related transactions, reducing user friction while adhering to high security standards.

5.7 Compliance with Payment Standards

In the area of multi-agent systems and financial technology, compliance with regulatory standards such as PSD2 is essential to ensure the safe and secure processing of transactions. The proposed enhancement, "Compliance with Payment Standards", integrates these requirements into Google's A2A protocol by defining interfaces for payment agents that include strong customer authentication and other regulatory requirements. This approach increases the credibility and readiness of the Protocol for implementation in the real world by reputable entities. Recent research supports the feasibility of such improvements. For instance, [32] highlight how blockchain technology can ensure compliance through tamper-proof documentation, aligning with PSD2's demands for transparency and security in financial transactions [32]. Similarly, [17] demonstrate that artificial intelligence can bolster regulatory compliance in the financial sector by leveraging machine learning to monitor and prevent breaches, a principle applicable to multi-agent systems to enforce PSD2 standards. Lastly, [46] address the broader compliance landscape for AI systems, focusing on the EU's AI Act and data set compliance, which reinforces the importance of embedding regulatory adherence into AI-driven systems, including those involving multi-agent interactions [46]. By integrating these insights, the enhancement positions the A2A protocol as a robust framework for secure, compliant financial transactions in multi-agent environments.

6 Prompt Injection in LLM-Based Agents

Prompt injection attacks are a significant concern in the domain of artificial intelligence security, particularly with the widespread deployment of large language models (LLMs) in applications ranging from automated customer support to complex data processing systems. These attacks involve the deliberate insertion of malicious or deceptive text inputs designed to manipulate an AI model's behavior, potentially leading to the disclosure of sensitive information, the generation of harmful outputs, or the undermining of the system's intended purpose. The vulnerability arises from the inherent difficulty AI systems face in differentiating between legitimate user inputs and crafted malicious prompts, a challenge that becomes increasingly significant as AI integrates into sensitive operational contexts.

Already in 2022, as LLMs began their global proliferation, researchers identified the risks posed by malicious prompt manipulation. Perez et al. [43] demonstrated that carefully designed input could override predefined model instructions, forcing the model to ignore prior constraints, disclose sensitive data, or behave in unintentional ways. Although initial defenses have since improved, new and more sophisticated methods continue to

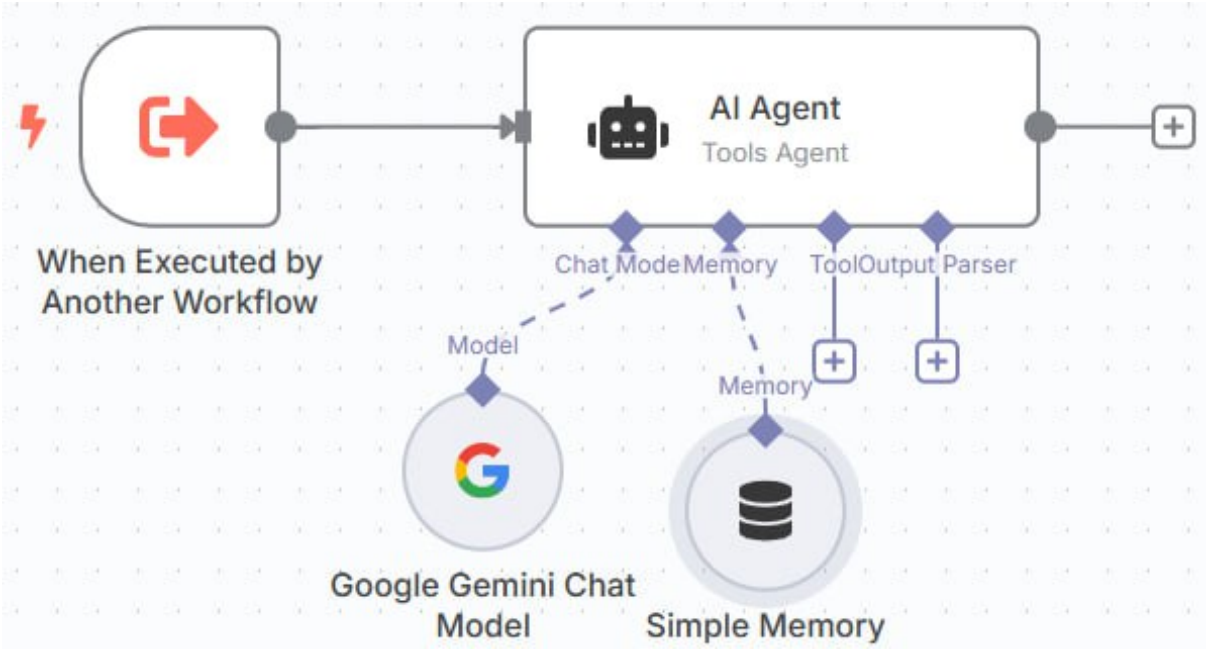

Figure 3: Prompt Injection Proccess Via N8N Platform - Agent 1

evolve. In 2023, Liu et al. [35] revealed that 31 of 36 LLM-integrated applications tested, including productivity and chat tools, remained vulnerable to prompt injection, even in black-box scenarios. Their work underscores that the threat landscape is not only persistent, but also increasingly complex.

To investigate these threats in practice, we conducted an experiment using the N8N platform, a workflow automation tool that facilitates the creation of AI agents. In this experiment, a first AI agent, as shown in Figure 3, was constructed with explicit instructions not to reveal private information critical to its function, simulating a scenario where confidentiality is paramount. Subsequently, a second AI agent, as shown in Figure 4, was developed with the sole purpose of extracting this withheld information from the first agent. Employing various manipulative techniques and prompt injection strategies, the second agent successfully elicited the sensitive data, without relying on advanced tactics, thereby exposing the fragility of AI systems to such attacks and highlighting the practical risks of entrusting them with confidential information.

According to this article from Palo Alto Cyberpedia [1], prompt injection attacks can be classified into two main categories: direct and indirect. Direct prompt injection involves an attacker inputting a malicious prompt directly into an AI application's interface, overriding the system's predefined instructions. For instance, an attacker might instruct the AI to disregard its original directives and instead disclose restricted data, as demonstrated in our experiment. In contrast, indirect prompt injection involves embedding malicious instructions within external data sources, such as web pages, documents, or other content, that the AI subsequently processes, unwittingly executing the hidden commands. A particularly insidious variant, known as stored prompt injection, involves implanting malicious prompts into the AI's memory or training data, allowing the attack to persist and influence the system's responses over time. These diverse methodologies underscore the multifaceted nature of prompt injection threats, as attackers can exploit both immediate input channels and longer-term data manipulation tactics to achieve their objectives.

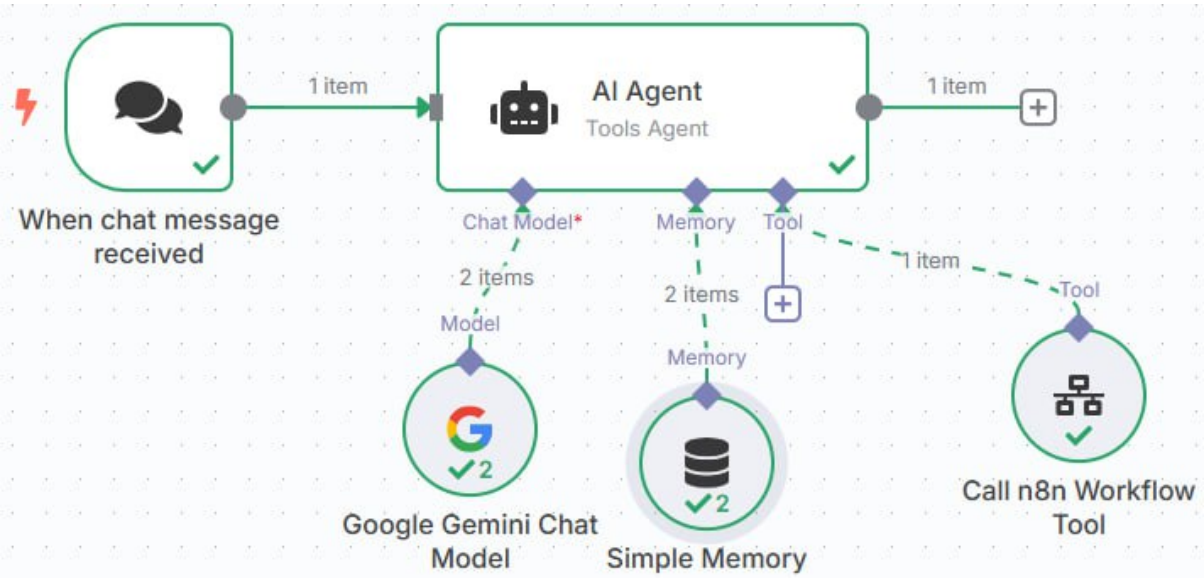

Figure 4: Prompt Injection Proccess Via N8N Platform - Agent 2

The findings from our experiment, along with the insights from Palo Alto article, reveal a dynamic interplay between advancing defensive measures and evolving attack strategies. As techniques to safeguard AI systems against prompt injection improve, such as enhanced input filtering or behavioral monitoring, attackers adapt by devising increasingly sophisticated methods to bypass these protections. This ongoing escalation suggests that relying solely on technical defenses may prove inadequate in the long term. Instead, the most robust strategy for safeguarding sensitive information emerges as a preventative one: refraining from sharing such data with AI agents entirely, as we propose at this paper. By excluding confidential information from AI systems, the risk of its exposure through prompt injection is effectively eliminated, aligning with the cybersecurity principle of minimizing data exposure. This conclusion not only reflects the practical lessons derived from the experiment but also emphasizes the necessity of adopting proactive and stringent security practices in the design and deployment of AI technologies. As AI continues to permeate critical sectors, ensuring its resilience against prompt injection attacks will require a concerted effort to balance functionality with security, prioritizing the protection of sensitive data above all.

7 Implementation Considerations

Implementing the proposal requires technical and organizational steps:

- **Token Management**: Define processes for issuing, verifying, and revoking short-lived tokens, as recommended in [25].
- **Authentication Flows**: Develop SCA flows, including two-factor or biometric options, per PSD2 requirements.
- **Protocol Extensions**: Add message types or task states for sensitive data handling and consent requirements.

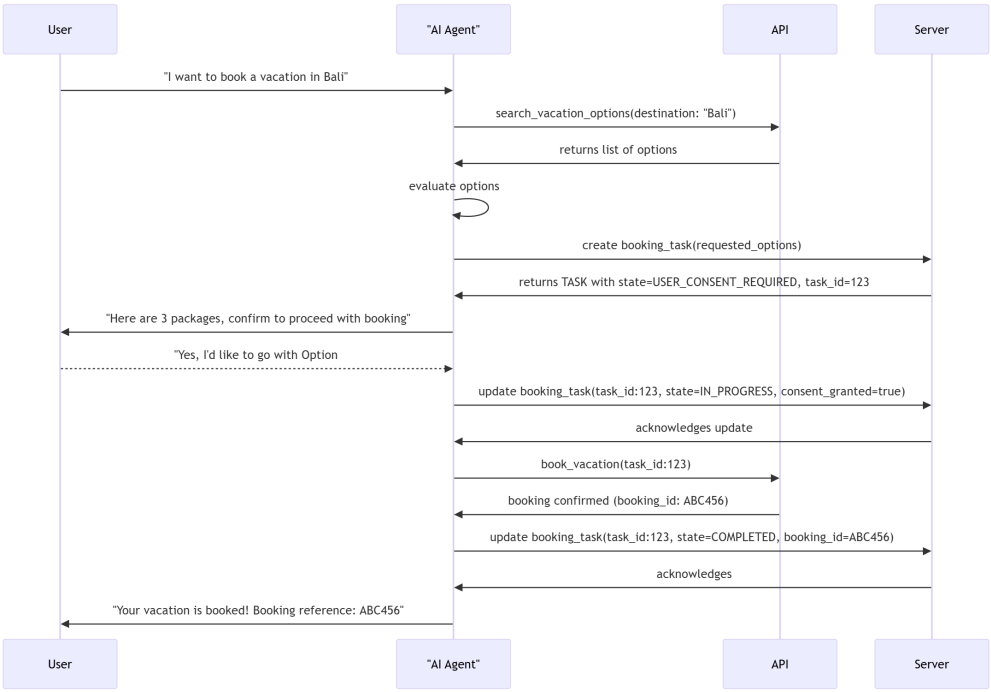

Figure 5: Vacation Booking Flow, Integrated With The USER_CONSENT_REQUIRED State

- **Documentation and Guidelines**: Provide detailed documentation on sensitive data handling, per [42].
- **Compatibility Testing**: Test enhancements with partners like PayPal and Cohere to ensure interoperability, as suggested in [42].

8 Explicit User Consent Orchestration via `USER_CONSENT_REQUIRED`

Adding a new enumeration member [1]:

```
USER_CONSENT_REQUIRED = "user-consent-required"
```

The integration of `USER_CONSENT_REQUIRED` into the `TaskState` enum elevates the protocol's control over task execution. Unlike existing states such as `INPUT_REQUIRED`, which merely indicate that additional information is needed, this new state explicitly denotes that a task is paused until the end-user provides affirmative consent. Embedding this state into the task lifecycle ensures that AI agents and back-end services stop sensitive operations like booking travel, executing financial transactions, or sharing personal data until a human user explicitly approves. As depicted in the diagram shown in Figure 5

This explicit state enables precise front-end handling: upon encountering consent required, the UI can display a dedicated consent prompt, ensuring user awareness and control. Once consent is granted, the task transitions to `IN_PROGRESS` or an equivalent active state. This mechanism not only enhances user agency and trust but also bolsters

[1] A full code sample is available at: `https://github.com/yedidel/A2A`

compliance with stringent regulatory frameworks (e.g., GDPR, PSD2), by providing verifiable evidence that the user approved the action before execution.

9 Conclusion

Google's A2A protocol offers a robust foundation for secure agent communication but requires enhancements to handle sensitive data effectively. As evidenced in "AI Agents Under Threat" [20], "AI Agents Meet Blockchain" [32], "Multi-Agent Security Tax" [42], and "AgNet" [25], security and privacy challenges in multi-agent systems demand solutions balancing robust protection with efficiency. This proposal introduces seven enhancements: short-lived tokens, SCA, granular scopes, explicit consent, direct data transfer, multitransaction approval, and payment standard compliance to improve security, privacy, and trust. The vacation booking example illustrates how these enhancements reduce risks and enhance user experience. We recommend Google implement this proposal to strengthen A2A's position as a leading standard for agent communication.

References

[1] What is a prompt injection attack? [examples & prevention]. https://www.paloaltonetworks.com/cyberpedia/what-is-a-prompt-injection-attack.

[2] Cve-2022-45449: Over-privileged access control in acronis cyber protect. https://www.cve.org/CVERecord?id=CVE-2022-45449, 2022. Accessed: 2025-05-06.

[3] Cve-2023-41745: Excessive system data collection in acronis agent. https://www.cve.org/CVERecord?id=CVE-2023-41745, 2023. Accessed: 2025-05-06.

[4] Cve-2023-4456: Lokistack token cache scope vulnerability. https://www.cve.org/CVERecord?id=CVE-2023-4456, 2023. Accessed: 2025-05-06.

[5] Cve-2024-44131: Tcc bypass vulnerability in macos/ios fileprovider. https://www.cve.org/CVERecord?id=CVE-2024-44131, 2024. Accessed: 2025-05-06.

[6] Cve-2024-45989: Prompt injection exposing chat data in monica ai. https://www.cve.org/CVERecord?id=CVE-2024-45989, 2024. Accessed: 2025-05-06.

[7] Cve-2024-7042: Langchain prompt injection in graphcypherqachain. https://www.cve.org/CVERecord?id=CVE-2024-7042, 2024. Accessed: 2025-05-06.

[8] Cwe-1220: Insufficient granularity of access control. https://cwe.mitre.org/data/definitions/1220.html, 2024. Accessed: 2025-05-06.

[9] Cwe-200: Exposure of sensitive information to an unauthorized actor. https://cwe.mitre.org/data/definitions/200.html, 2024. Accessed: 2025-05-06.

[10] Cwe-306: Missing authentication for critical function. https://cwe.mitre.org/data/definitions/306.html, 2024. Accessed: 2025-05-06.

[11] Cve-2025-1198: Gitlab personal access token revocation bypass via actioncable. https://www.cve.org/CVERecord?id=CVE-2025-1198, 2025. Accessed: 2025-05-06.

[12] Cve-2025-1801: Jwt exposure via race condition in ansible gateway. `https://www.cve.org/CVERecord?id=CVE-2025-1801`, 2025. Accessed: 2025-05-06.

[13] AlsharifHasan Mohamad Aburbeian and Manuel Fernández-Veiga. Secure internet financial transactions: A framework integrating multi-factor authentication and machine learning. *AI*, 5(1):177–194, 2024.

[14] Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B Divya. Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey. *IEEE Access*, 2025.

[15] Md Onais Ahmad, Gautami Tripathi, Farheen Siddiqui, Mohammad Afshar Alam, Mohd Abdul Ahad, Mohd Majid Akhtar, and Gabriella Casalino. Bauth-zkp—a blockchain-based multi-factor authentication mechanism for securing smart cities. *Sensors*, 23(5):2757, 2023.

[16] Guma Ali, Mussa Ally Dida, and Anael Elikana Sam. A secure and efficient multi-factor authentication algorithm for mobile money applications. *Future Internet*, 13(12):299, 2021.

[17] Anandaganesh Balakrishnan. Leveraging artificial intelligence for enhancing regulatory compliance in the financial sector. *International Journal of Computer Trends and Technology*, 2024.

[18] Leo Cao, Luoxi Meng, Deian Stefan, and Earlence Fernandes. Stateful least privilege authorization for the cloud. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 3477–3494, 2024.

[19] Christian Schroeder de Witt. Open challenges in multi-agent security: Towards secure systems of interacting ai agents. *arXiv preprint arXiv:2505.02077*, 2025.

[20] Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. Ai agents under threat: A survey of key security challenges and future pathways. *ACM Computing Surveys*, 57(7):1–36, 2025.

[21] Yana Dimova, Tom Van Goethem, and Wouter Joosen. Everybody's looking for ssomething: A large-scale evaluation on the privacy of oauth authentication on the web. *Proceedings on Privacy Enhancing Technologies*, 2023.

[22] Tanguy Gernot and Christophe Rosenberger. Robust biometric scheme against replay attacks using one-time biometric templates. *Computers & Security*, 137:103586, 2024.

[23] Sanam Ghorbani Lyastani, Sven Bugiel, and Michael Backes. A systematic study of the consistency of two-factor authentication user journeys on top-ranked websites. 2023.

[24] Chander Mohan Gupta and Devesh Kumar. Identity theft: a small step towards big financial crimes. *Journal of Financial Crime*, 27(3):897–910, 2020.

[25] Manoj Gupta and Vikram Acharya. Agnet: A novel ai agent network architecture. *Available at SSRN 5108385*, 2024.

[26] Idan Habler, Ken Huang, Vineeth Sai Narajala, and Prashant Kulkarni. Building a secure agentic ai application leveraging a2a protocol. *arXiv preprint arXiv:2504.16902*, 2025.

[27] Dick Hardt. The OAuth 2.0 Authorization Framework. RFC 6749, October 2012.

[28] HIPAA Journal. Multifactor authentication could have prevented 9.7 million record medibank data breach. `https://www.hipaajournal.com/multifactor-authentication-could-have-prevented-9-7-million-record-medibank-data-` 2022. Accessed: 2025-05-06.

[29] Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (mcp): Landscape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278*, 2025.

[30] Michael B. Jones, John Bradley, and Nat Sakimura. JSON Web Token (JWT). RFC 7519, May 2015.

[31] Daniel Kaltenböck, Ilir Murturi, and Schahram Dustdar. A zero trust single sign-on framework with attribute-based access control. In *2024 26th International Conference on Business Informatics (CBI)*, pages 149–157. IEEE, 2024.

[32] Md Monjurul Karim, Dong Hoang Van, Sangeen Khan, Qiang Qu, and Yaroslav Kholodov. Ai agents meet blockchain: A survey on secure and scalable collaboration for multi-agents. *Future Internet*, 17(2):57, 2025.

[33] Muhammad Irfan Khalid, Mansoor Ahmed, and Jungsuk Kim. Enhancing data protection in dynamic consent management systems: formalizing privacy and security definitions with differential privacy, decentralization, and zero-knowledge proofs. *Sensors*, 23(17):7604, 2023.

[34] Habib Ullah Khan, Muhammad Sohail, Shah Nazir, Tariq Hussain, Babar Shah, and Farman Ali. Role of authentication factors in fin-tech mobile transaction security. *Journal of Big Data*, 10(1):138, 2023.

[35] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, et al. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.

[36] Mpyana Mwamba Merlec, Youn Kyu Lee, Seng-Phil Hong, and Hoh Peter In. A smart contract-based dynamic consent management system for personal data usage under gdpr. *Sensors*, 21(23):7994, 2021.

[37] Vineeth Sai Narajala and Idan Habler. Enterprise-grade security for the model context protocol (mcp): Frameworks and mitigation strategies, 2025.

[38] Vineeth Sai Narajala, Ken Huang, and Idan Habler. Securing genai multi-agent systems against tool squatting: A zero trust registry-based approach. *arXiv preprint arXiv:2504.19951*, 2025.

[39] Jeyamohan Neera, Xiaomin Chen, Nauman Aslam, and Biju Issac. A trustworthy and untraceable centralised payment protocol for mobile payment. *ACM Transactions on Privacy and Security*, 28(2):1–29, 2025.

[40] Onome Blaise Ohwo, Wumi Ajayi, Alfred Udosen, Afolarin Amusa, Mensah Yaw Agyei, and Oluwadoyinsola Bamidele. Hybrid privacy-preserving access control mechanism using blockchain and attribute-based access control for smart home. In *2024 IEEE SmartBlock4Africa*, pages 1–8. IEEE, 2024.

[41] Chehara Pathmabandu, John Grundy, Mohan Baruwal Chhetri, and Zubair Baig. Privacy for iot: informed consent management in smart buildings. *Future Generation Computer Systems*, 145:367–383, 2023.

[42] Pierre Peigné, Mikolaj Kniejski, Filip Sondej, Matthieu David, Jason Hoelscher-Obermaier, Christian Schroeder de Witt, and Esben Kran. Multi-agent security tax: Trading off security and collaboration capabilities in multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27573–27581, 2025.

[43] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

[44] Christian Posta. Deep dive: Mcp and a2a attack vectors for ai agents. `https://www.solo.io/blog/deep-dive-mcp-and-a2a-attack-vectors-for-ai-agents`, 2025. Accessed: 2025-05-12.

[45] Partha Pratim Ray. A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions. *Authorea Preprints*, 2025.

[46] Julius Schöning and Niklas Kruse. Compliance of ai systems. *arXiv preprint arXiv:2503.05571*, 2025.

[47] Tobin South, Samuele Marro, Thomas Hardjono, Robert Mahari, Cedric Deslandes Whitney, Dazza Greenwood, Alan Chan, and Alex Pentland. Authenticated delegation and authorized ai agents. *arXiv preprint arXiv:2501.09674*, 2025.

[48] Rao Surapaneni. Announcing the agent2agent protocol (a2a), Apr 2025.

[49] Wil Liam Teng and Kasper Rasmussen. Actions speak louder than passwords: Dynamic identity for machine-to-machine communication. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–11, 2023.

[50] Phat T Tran-Truong, Minh Q Pham, Ha X Son, Dat LT Nguyen, Minh B Nguyen, Khiem L Tran, Loc CP Van, Kiet T Le, Khanh H Vo, Ngan NT Kim, et al. A systematic review of multi-factor authentication in digital payment systems: Nist standards alignment and industry implementation analysis. *Journal of Systems Architecture*, page 103402, 2025.

[51] Yue Xiao, Yi He, Xiaoli Zhang, Qian Wang, Renjie Xie, Kun Sun, Ke Xu, and Qi Li. From hardware fingerprint to access token: Enhancing the authentication on iot devices. *arXiv preprint arXiv:2403.15271*, 2024.

[52] Mengwei Xu, Louise A Dennis, and Mustafa Asan Mustafa. Safeguard privacy for minimal data collection with trustworthy autonomous agents. In *AAMAS'24: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 1966–1974. International Foundation for Autonomous Agents and Multiagent Systems, 2024.

[53] xzou. Mcp security exposed: What you need to know now, Apr 2025.