Closing Early Attack Surfaces in Bluetooth Secure Simple Pairing using Identity-Based Signatures *

Woori Bae and Taek-Young Youn†

Dankook University, Yongin, Republic of Korea
{dnfl0117, taekyoung}@dankook.ac.kr

Abstract

Bluetooth BR/EDR Secure Simple Pairing (SSP) can be exposed in its early phase to active attacks such as confusion, downgrade, and key substitution. The TOFU-or-DOFU model strengthens past/future key security via deferrable authentication, but it does not structurally preclude the attacks before deferrable authentication. This paper proposes IBS-based SSP, which inserts an ID-Based Signature (IBS) check immediately after the SSP Key exchange. Using BD_ADDR as the ID, it binds g^a, g^b, Role, the IOcap/AuthReq summary $FlagSet$, and the selected model $MethodTag$ into a single signature input, blocking the attacks as Role Confusion, Method Confusion, Pairing Confusion, Ghost Keystroke, and Downgrade to Just Works. The overhead is limited to a $\sigma \approx 96B$ signature (two compressed points at 128-bit security), carried within the standard LMP Encapsulated PDU, making it more efficient than certificate-based alternatives.

1 Introduction

Bluetooth, introduced in the 1990s, has become the standard for short-range wireless communication and is used across a wide range of devices. As Bluetooth has proliferated, attacks targeting it have increased, creating the need to ensure data confidentiality, integrity, and mutual authentication between devices. Bluetooth communication is divided into BR/EDR (Basic Rate/Enhanced Data Rate) and LE (Low Energy). BR/EDR is mainly used in high-bandwidth, low-latency applications such as audio and automotive, while LE is primarily used in IoT and low-power scenarios. This paper focuses on BR/EDR.

BR/EDR uses Legacy Pairing and Secure Simple Pairing (SSP) to generate a Link Key (LK) between two devices attempting to connect. Legacy Pairing derives the LK using the SAFER+-based E21 or E22 algorithms. Secure Simple Pairing (SSP) uses FIPS-approved algorithms such as SHA-256, HMAC-SHA-256, and ECDH. Compared to Legacy Pairing, SSP employs stronger algorithms and is therefore better suited to modern security requirements.

However, during the early phase of the SSP procedure, attacks may occur that exploit incomplete mutual authentication or attempt to tamper with data during pairing. A representative example is the Ghost Keystroke attack, which leverages the fact that the keys for LK generation and user inputs such as Passkey Entry are not strongly bound to the session, thereby inducing the user to establish the wrong key with an unintended peer. In addition, attacks such as IOcap Downgrade, Role Confusion, and Method Confusion can also prevent correct key generation under SSP.

As an approach to address these vulnerabilities, the Trust on First Use or Deferrable Authentication (TOFU-or-DOFU) security model has recently been proposed. In this model, the

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 70, December 16-18, 2025, Sapporo , Japan.

†Corresponding author

LK derived in the first connection is kept as is (TOFU), and when the user later succeeds in a signature-based challenge–response using certificates (DOFU), the freshness of future keys as well as the security of past keys are guaranteed (TD Pairing).

However, this model does not structurally block active attacks before authentication. If an attacker interfered early, deferrable authentication can only prove that the peer is legitimate—or detect that it is not and recover by re-pairing and regenerating keys. It cannot protect past sessions that were exposed before authentication.

Therefore, to overcome this limitation, this paper proposes ID-Based Signature based Secure Simple Pairing (IBS-based SSP), which enables omission of the deferrable authentication process suggested in the TOFU-or-DOFU model by structurally blocking attacks in the initial pairing phase of BR/EDR. At the DH key exchange point targeted by attacks in the initial pairing phase, an ID-Based Signature (IBS) is exchanged and verified; if an attacker tampers with the pairing process, signature verification fails and the attack is structurally blocked. In the proposed method, no certificates are required: each device holds an ID-based private key issued in advance by a Private Key Generator (PKG), and verification is performed on the transmitted signature by reusing the standard key-exchange packet, the LMP Encapsulated PDU.

2 Background

2.1 Secure Simple Pairing

Secure Simple Pairing (SSP) is the BR/EDR procedure for generating an LK between two devices for connection establishment, and it is divided into four stages in the standard: (1) Public Key Exchange, (2) Authentication Stage 1, (3) Authentication Stage 2, and (4) Link Key Calculation. The overall flow of SSP is shown in Figure 2, and the details of each stage are as follows.[3]

(1) **Diffie–Hellman Key Exchange:** The initiator device A and responder device B exchange public values to compute g^{ab}: device A first sends its key g^a, and B responds with its key g^b.

(2) **Authentication Stage 1:** In this stage, depending on each side's IO capability (IOcap) and support for a secure channel, one of Numeric Comparison (NC), Out-of-Band (OOB), or Passkey Entry (PKE) is selected. Just Works uses the same protocol as NC but omits user authentication.

- **Numeric Comparison (NC):** Each side generates a 128-bit nonce. B computes and sends a commitment using g^a, g^b, and B's nonce; the parties then exchange their nonce. A recomputes the same commitment for verification and derives a 6-digit check value for on-screen comparison on both devices.
- **Out-of-Band (OOB):** The OOB model is used only when the OOB authentication information reception flag is set. If bi-directional OOB is available, mutual authentication is performed using commitments exchanged over OOB; with uni-directional OOB, B proves knowledge of the secret received via OOB. If pairing begins with OOB, the key exchange occurs after the OOB step.
- **Passkey Entry (PKE):** PKE is used when designated during the IOcap exchange. The same 6-digit (20-bit) passkey is entered on both devices. For each bit of the passkey, the parties exchange nonce-based (128-bit) commitments and reveal them

sequentially; on any mismatch, the procedure aborts immediately and subsequent bits are not disclosed. Figure 2 shows the SSP flow under PKE. Specifically, for PKE, per-bit nonces $N_{a,i}$ and $N_{b,i}$ are used to compute commitments C_{a_i} and C_{b_i}, which are exchanged, followed by nonce exchange and verification for each bit.

(3) **Authentication Stage 2:** This final mutual-authentication step is common to all three models (NC/OOB/PKE). Each side computes a confirmation value from the previously exchanged values and the newly agreed shared key, and they verify each other's value. If either device fails verification, pairing is aborted immediately.

(4) **Link Key Calculation:** The LK is derived from the agreed DHKey together with the exchanged nonces and device addresses. To guarantee a correct LK, both devices must use exactly the same parameter ordering. The resulting LK is then used as the root key for maintaining the connection and for deriving higher-layer keys.

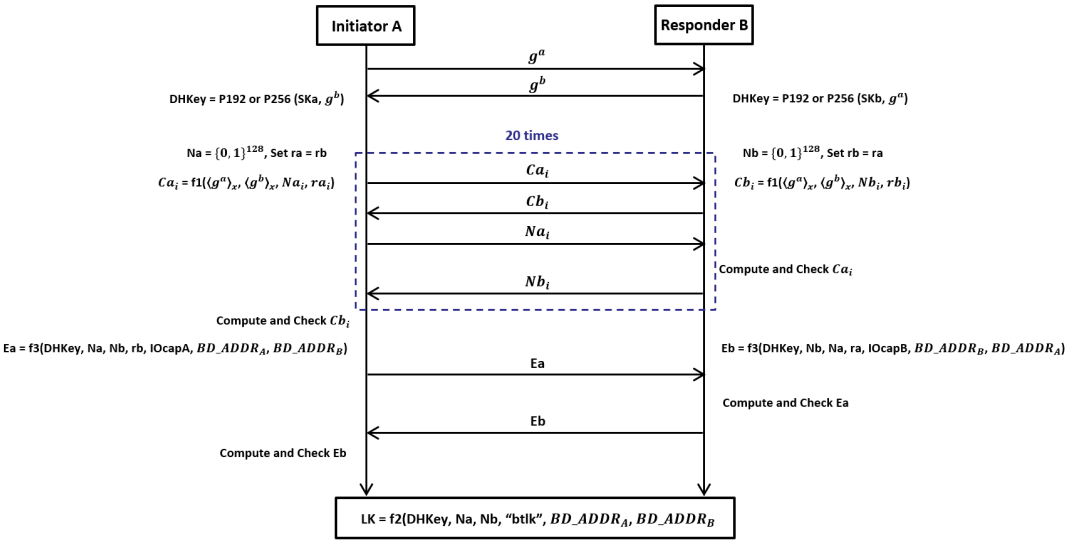

Figure 1: Secure Simple Pairing: Passkey Entry

2.2 TOFU-or-DOFU Security Model

2.2.1 Trust on First Use(TOFU)

The TOFU model formalizes, in a game-based model , the actual BR/EDR operational flow consisting of (i) key agreement during the initial pairing, (ii) session-key derivation upon subsequent reconnections, and (iii) optional authentication. In Bluetooth, once initial pairing is completed, only the device identifier, the LK, and the locally stored DHKey are used for reconnection and authentication. Consequently, the TOFU model reflects this practical constraint by separating initial pairing, reconnection, and authentication into distinct sessions, allowing the adversary to arbitrarily schedule attacks across these times.

Two core security requirements are fundamentally demanded in this model: match security and key secrecy. Their definitions are as follows:

- **Match Security**: Even in the presence of concurrent executions and network delays, only two executions whose session identifier (defined from the public parameters of the initial connection) matches recognize each other as partners and derive the same link key. Thus, even if an adversary intervenes in various ways, differing session identifiers make the probability of sharing the same key negligible.

- **Key Secrecy**: The session key retains computational secrecy. In TOFU, the predicate `isTOFU`—meaning there was no active attack during the initial connection—is the admissibility condition for a session. Only session keys derived under `isTOFU` = ON are subject to the adversary's test, which yields indistinguishability and hence secrecy. Conversely, if `isTOFU` = OFF, an active attack may have occurred and the key might already be exposed; such sessions are excluded from the security target.

However, TOFU starts from the assumption of no active attacker and, due to the possible reuse of the LK and other keys in Bluetooth, does not provide strong forward secrecy. In other words, connections in which active attacks (e.g., Method Confusion, Role Confusion) occur during the initial pairing phase remain outside the TOFU model's security coverage.[1]

2.2.2 TOFU-or-DOFU Security Model

The TOFU-or-DOFU security model extends the formal framework of TOFU by combining initial key agreement (TOFU) and deferrable authentication (DOFU) in the SSP of Bluetooth Secure Connections. The core idea is to generate an LK through initial pairing without changing the standard stack's flow, and then allow the user to additionally perform a certificate-based challenge–response at a desired time. If this deferrable authentication succeeds even once, it guarantees not only the security of future session keys but also retroactively the security of keys derived earlier. In other words, even if an attacker interfered during the initial connection, a later successful authentication can substantially reduce the attack surface across both past and future sessions. Under this premise, the TOFU-or-DOFU security model can block various active attacks.

Formally, while preserving TOFU's two security properties—key secrecy and match security—the model introduces, in addition to `isTOFU`, a flag `isPartnerAuth` indicating whether the peer has completed authentication. Even if there was active interference at the beginning, once deferrable authentication for that connection succeeds and `isPartnerAuth` becomes ON, the keys derived from that connection become admissible for testing and key secrecy is guaranteed. Intuitively, the authentication stage retroactively establishes the legitimacy of the initial connection; the adversary cannot test keys of connections interfered with before authentication, and even after authentication, must demonstrate that the legitimate partner actually holds the same LK. The model assumes unilateral authentication by default, and running the same procedure in the opposite direction upgrades it to mutual authentication. In addition, session identifiers are clearly fixed per stage to maintain match security. Consequently, TOFU-or-DOFU replaces TOFU's premise of "secure only if no early active attacker" with a guarantee that, even if there was early active interference, once deferrable authentication succeeds, both past and future session keys are secure from that point onward.[1]

2.3 Existing Attacks

This section summarizes active attacks targeting the SSP of BR/EDR. The threat model assumes a man-in-the-middle adversary who exploits structural gaps to perform modification or injection between the two devices. We select five representative attacks—Role Confusion, Method Confusion, Pairing Confusion, Ghost Keystroke, and Downgrade to Just Works—and describe them as follows:

- **Role Confusion**: The attacker tampers with role-related packets exchanged between the Initiator and the Responder, causing each device to misidentify its own role. As a result, the role mismatch undermines the assumptions of the authentication procedure.[5]
- **Method Confusion**: By modifying the IOcap exchange, the attacker makes the two devices select different association methods. For example, one device performs NC while the other performs PKE, producing a user experience that appears similar. Consequently, the intended binding is bypassed and the attack succeeds.[6]
- **Pairing Confusion**: By altering exchanges such as IOcap and AuthReq, the attacker makes the two parties execute different BR/EDR pairing families within the same session. Concretely, one side is driven to Legacy Pairing while the other proceeds with SSP, forcing both devices to derive an LK of the attacker's choosing.[7]
- **Ghost Keystroke**: This attack occurs in the PKE model. The attacker first establishes a separate connection with the input device and induces the creation of a link key LK_{comp} using the attacker's DH public value. When the user attempts to connect to the output device, the attacker lures the user into entering the on-screen passkey on the input device, collects the keystrokes over the pre-existing connection, impersonates the input device to the output device, and completes pairing so that a desired key is established.[8]
- **Downgrade to Just Works**: By tampering with the IOcap exchange, the attacker steers the devices into the non-authenticating Just Works model, effectively skipping the intended authentication. Although pairing appears normal to the user, key agreement is completed without authentication, significantly increasing the risk of man-in-the-middle attacks.[9]

2.4 ID-Based Signature

An ID-Based Signature (IBS) is a signature scheme that uses the identifier itself as the public key without generating an additional public key. And a Private Key Generator (PKG) issues a private key corresponding to each identifier from a master secret. In this setting, the verifier validates a signature using a curve point mapped from the identifier, e.g., $Q_{ID} = H_2(ID)$, without any certificate verification. This makes IBS useful in environments that require lightweight procedures. The IBS construction used in this paper operates under the Gap Diffie–Hellman (GDH) assumption, under which its security and verification equation are defined. The detail process of IBS consists of Setup, Extract, Sign and Verify.[2]

IBS signature size depends on the size of group elements. Using a single (compressed) coordinate as the unit, the per-point size by security level is summarized as follows:

- 128-bit (BLS12-381) $\approx$ 48 bytes
- 192-bit (BLS24-479) $\approx$ 60 bytes
- 256-bit (BLS48-581) $\approx$ 73 bytes

3 Our proposed Protocol

3.1 Basic Idea

The existing TOFU-or-DOFU security model guarantees the security of keys for both past and future sessions through certificate-based challenge–response. However, if post-hoc authentication fails, one can regenerate the LK to reestablish the security of keys generated in the future, but there is no remedy for attacks that occurred in the past. Therefore, this paper proposes a security model that replaces the TOFU component within TOFU-or-DOFU and a protocol that provides security from the initial session prior to DOFU, thereby obviating the need for DOFU. Specifically, we propose ID-Based Signature Based Secure Simple Pairing (IBS-based SSP), which structurally blocks active attacks from the outset by additionally exchanging and verifying an ID-Based Signature during the initial key exchange of SSP. Here, the ID is the Bluetooth device address BD_ADDR itself. This allows us to reuse the address already shared during device discovery, requiring no extra identifier exchange. During the initial key exchange for DHKey generation, the parties additionally exchange an IBS and verify it using ID-based private keys issued in advance by a Private Key Generator (PKG). Keys and system parameters issued by the PKG are pre-provisioned into each device during the Bluetooth qualification process.

3.2 ID-Based Signature Based Secure Simple Pairing

In this paper, we propose ID-Based Signature Based Secure Simple Pairing (IBS-SSP), which inserts an ID-based signature verification immediately after the DH key exchange of the standard BR/EDR SSP so that the exchanged g^a, g^b and the essential data for that exchange are bound by a single signature. The ID used in IBS is the device identifier BD_ADDR itself without any additional exchange, and we assume that the private key D_ID issued by the PKG to each device and the public parameters (P, P_{pub}, H_1, H_2) are pre-provisioned into devices at manufacturing time. Thus, while preserving the structure of standard SSP, we preemptively block early active attack surfaces at the signature stage.

The core flow of our protocol is shown in Figure 3, and the procedure for signature generation is as follows.

1. **Key generation**: Initiator A and Responder B generate the DH public values g^a, g^b for DHKey derivation as in SSP.
2. **ID point computation**: Each device computes $Q_{ID\_A} = H_2(BD\_ADDR\_A)$ and $Q_{ID\_B} = H_2(BD\_ADDR\_B)$ from its device address.
3. **Flag dataset construction**: Summarize the IOCap and Authreq flags exchanged at the start of SSP:

   $$FlagSet := H(IOcap\_A \parallel IOcap\_B \parallel AuthReq\_A \parallel AuthReq\_B)$$

   and denote the association model determined from IOCap and Authreq as a 1-byte $MethodTag \in \{NC, PKE, OOB, JW\}$.
4. **Signature-input construction**:
   - For $A \rightarrow B$, A builds

     $$m_A = BD\_ADDR\_A \parallel BD\_ADDR\_B \parallel Role\_A(\text{Initiator}) \parallel MethodTag \parallel FlagSet \parallel g^a.$$

- For $B \rightarrow A$, B builds

$$m_B = BD\_ADDR\_A \parallel BD\_ADDR\_B \parallel Role\_B(\text{Responder}) \parallel MethodTag \parallel FlagSet \parallel g^b.$$

5. **IBS signature generation**: Each device samples r, computes $U = rQ_{ID}$ and $h = H_1(m, U)$, and outputs $V = (r + h)D_{ID}$, i.e., $\sigma = (U, V)$. Concretely,

$$\sigma_A = (U_A, V_A), \quad U_A = r_A Q_{ID\_A},\ h_A = H_1(m_A, U_A),\ V_A = (r_A + h_A)D_{ID\_A},$$

$$\sigma_B = (U_B, V_B), \quad U_B = r_B Q_{ID\_B},\ h_B = H_1(m_B, U_B),\ V_B = (r_B + h_B)D_{ID\_B}.$$

   To minimize overhead, we use compressed G_1 points at 128-bit security on a BLS12-381 family curve, sending two points with $\sigma \approx 96$B.

6. **Transmission and verification**: A sends $g^a \parallel \sigma_A$. Then B computes Q_{ID_A} and h_A and checks

$$(P, P_{pub}, U_A + h_A Q_{ID\_A}, V_A) = (P, sP, (r_A + h_A)Q_{ID\_A}, s(r_A + h_A)Q_{ID\_A}).$$

   If verification fails, pairing aborts immediately. If it succeeds, B sends $g^b \parallel \sigma_B$, and A verifies in the same way. Only if both verifications pass do the parties proceed to the standard SSP Authentication stages 1 and 2 and to LK derivation.

Our protocol minimizes extra transmission to a single signature σ and binds already known $BD\_ADDR/Role/IOcap/AuthReq/MethodTag$ into the signature so that the verifier can reconstruct them without additional exchanges. Consequently, active interference during the key-exchange phase is blocked at the signature stage.

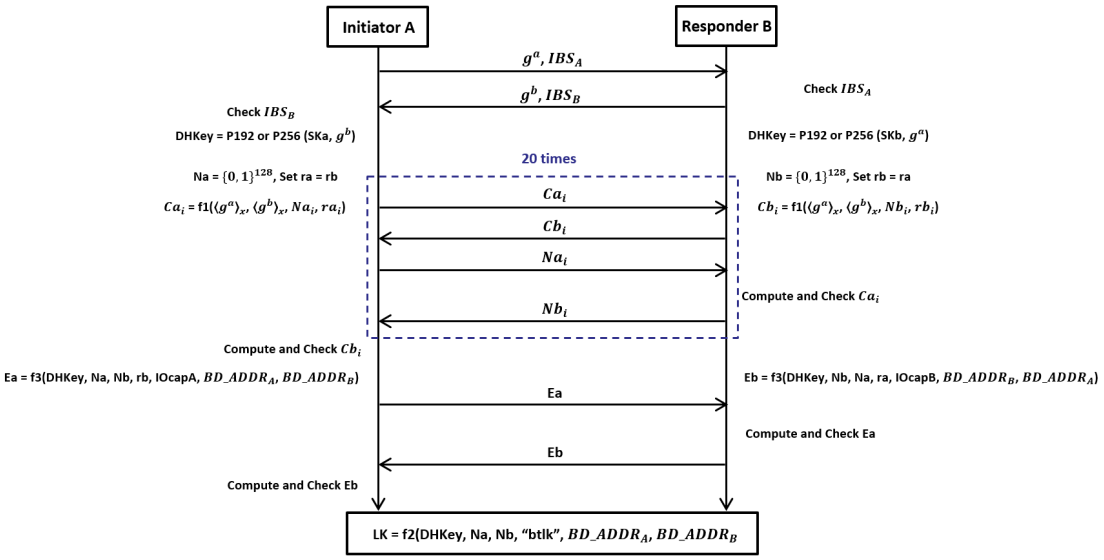

Figure 2: ID-Based Signature Based Secure Simple Pairing

4 Analysis

In this section, we present a security analysis of how the proposed IBS-based SSP blocks the five active attacks summarized in the Existing Attacks section, and we quantitatively evaluate the additional efficiency overhead under the transmission constraints of the BR/EDR standard. The analysis proceeds under the following assumptions: the PKG operates within a trusted manufacturing process and each device securely stores its internal D_{ID}. The adversary can eavesdrop on or modify packets over the Bluetooth link, but cannot break the security of IBS or exfiltrate D_{ID}.

4.1 Security Analysis against Existing Attacks

This section explains the mechanisms by which the proposed IBS-based SSP blocks the five active attacks defined in Existing Attacks (Role Confusion, Method Confusion, Pairing Confusion, Ghost Keystroke, Downgrade to JW).

- **Role Confusion:** The attacker attempts to confuse the initiator and responder roles between Bluetooth devices. Our message definition includes each device's role flags $Role_A, Role_B$; if the roles differ, the signature input itself changes and verification fails. Thus, Role Confusion cannot succeed without forging the IBS.

- **Method Confusion / Downgrade to JW:** These attacks manipulate IOcap and AuthReq so that the two devices select different association models (NC/PKE/OOB/JW) or are forced into JW with no authentication. Since the association model is deterministically chosen after the exchange of IOcap and AuthReq, we include in m_A and m_B the value $FlagSet = H(IOcap_A \parallel IOcap_B \parallel AuthReq_A \parallel AuthReq_B)$, which embeds both parties' IOcap and AuthReq. The selected model is also recorded as MethodTag. Therefore, if even a single bit is altered on one side, the two devices compute different $FlagSet$ values, the signature verification fails, and both attacks are structurally blocked.

- **Pairing Confusion:** The attacker makes the two devices execute different pairing families within the same session. In our protocol, even if one side is driven to Legacy Pairing and the other to SSP, the attacker cannot produce a valid IBS for the SSP side; the verification fails and the session aborts, preventing key forcing. Hence the goal of making both devices derive keys under attacker-chosen procedures is defeated.

- **Ghost Keystroke:** Ghost Keystroke is an attack in which the attacker interferes from the initial stage of the PKE model to force derivation of a desired key. The attacker attempts to send its own key early and establish a connection with the input device; however, our protocol requires the IBS exchange simultaneously with the initial key exchange, making it impossible for the attacker to establish such a connection with the input device. Consequently, the attacker cannot influence the key exchange to produce a chosen DHKey, and the attack is blocked.

4.2 Efficiency Analysis

This section analyzes the transmission overhead caused by the additional signature data, based on the 16-byte fragment format of the BR/EDR LMP Encapsulated PDU. We also compare the overhead with that of the existing TOFU-or-DOFU security model. To minimize overhead, our

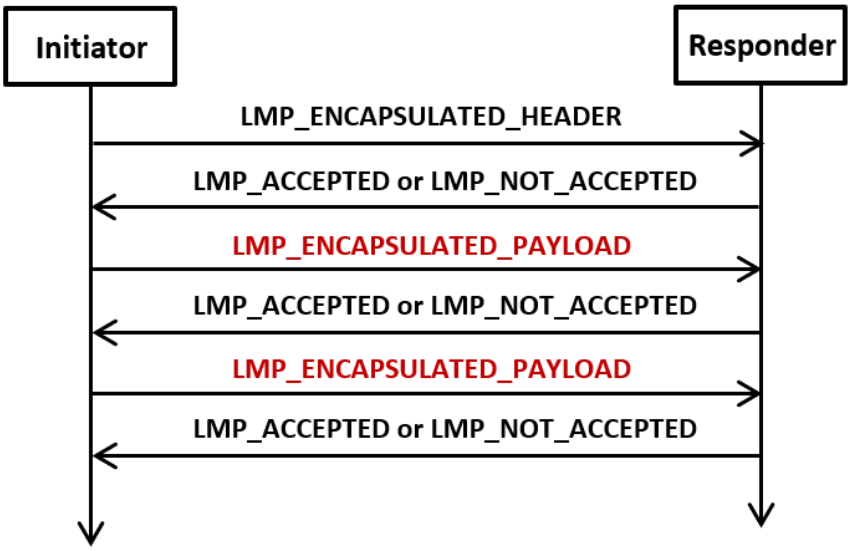

Figure 3: Encapsulated PDU

protocol fixes the IBS signature size at $\sigma \approx 96B$ at 128-bit security (two compressed points). We further assume an environment where P-256 keys are used for DHKey generation.

As shown in Figure 3, data exchange using the LMP Encapsulated PDU follows the standard's `LMP_ENCAPSULATED_HEADER` and LMP_ENCAPSULATED_PAYLOAD sequence. The HEADER indicates whether a Public Key is being sent, the key type (e.g., P-192 or P-256), and the total length of the forthcoming PAYLOAD; the Initiator sends this to the Responder. After the HEADER, LMP_ENCAPSULATED_PAYLOAD transmits data in 16-byte units, so the DH key material is segmented into 16-byte chunks and sent multiple times; for each chunk, the Responder replies with `LMP_ACCEPTED` or LMP_NOT_ACCEPTED. In the last PAYLOAD packet, any remaining bytes beyond the key material are padded with zeros.

Under this format, the standard SSP carries a P-256 public key (64 B) as four LMP_ENCAP SULATED_PAYLOAD packets (16B each) following the HEADER, with the Responder acknowledging each with LMP_ACCEPTED or LMP_NOT_ACCEPTED. When the IBS signature $\sigma \approx 96$ B is added, six additional PAYLOAD packets are required. Hence, per device, the signature introduces 6 (PAYLOAD) + 6 (ACK/NACK) = 12 extra packets; summing Initiator and Responder directions yields an overhead of about 24 packets per exchange.

In contrast, for the certificate-based alternative discussed in the TOFU-or-DOFU security model, an X.509 certificate can range from several hundred to several thousand bytes depending on extensions[4], resulting in a larger transmission overhead than IBS. Even considering compressed X.509 profiles proposed in prior work, the total additional transmission typically exceeds that of our 96 B signature. Therefore, the protocol proposed in this paper blocks early active attack surfaces during pairing by exchanging a compact signature with substantially lower overhead.

Table 1 reports the number of 16-byte LMP Encapsulated PAYLOAD fragments needed to carry a representative an example of X.509 certification(1520 B) versus our IBS signature (96 B). The certificate requires 95 fragments, whereas IBS needs only 6 fragments, i.e., a 93.7%

	Size of Example	Size of Packet	The Number of Fragment
X.509 Certification	1520B	16B	95
ID-Based Signature	96B	16B	6

Table 1: Efficiency Analysis

reduction (16/95) in fragment count. Since each fragment elicits one LMP ACCEPTED/NOT ACCEPTED reply on air, the per-device total on-air messages are 95(payload)+95(replies)=190 for X.509 and 6+6=12 for IBS—again a 93.7% reduction. Even with compressed X.509 profiles (e.g., 512 B $\rightarrow$ 32 fragments), transmissions remain 81.3% larger than IBS (6 fragments). In short, our protocol blocks early active attack surfaces by exchanging a compact signature while keeping the transmission overhead substantially lower.

5 Conclusion

This paper proposed IBS-based SSP, which combines an ID-Based Signature with the initial key exchange stage of BR/EDR Secure Simple Pairing (SSP). The proposed method uses the device identifier BD_ADDR as the ID and binds, into a single signature input, g^a, g^b, the role $Role$, the IOcap and Authreq flag dataset $FlagSet$, and the selected association model $MethodTag$.

Through this binding, the proposed protocol blocks five active attacks—Role Confusion, Method Confusion, Pairing Confusion, Ghost Keystroke, and Downgrade to JW. Because these attacks attempt to tamper with the data bound to the signature, they cannot succeed without forging the signature, allowing active attack surfaces to be eliminated from the very beginning of pairing.

At 128-bit security, the signature size for the two points $\sigma = (U, V)$ is minimized to $\sigma \approx 96$B. Transmission reuses the LMP Encapsulated PDU and carries only the additional signature during the key exchange. In contrast, certificate-based TOFU-or-DOFU approaches incur overhead proportional to certificate size. Consequently, our protocol preserves compatibility with the standard, minimizes potential overhead, and eliminates early-stage attack surfaces at the protocol level.

6 Acknowledgments

References

[1] Marc Fischlin, Olga Sanina. Fake It till You Make It: Enhancing Security of Bluetooth Secure Connections via Deferrable Authentication. CCS '24: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, Pages 4762 - 4776. https://doi.org/10.1145/3658644.3670360.

[2] Choon, J.C., Hee Cheon, J. (2003). An Identity-Based Signature from Gap Diffie-Hellman Groups. In: Desmedt, Y.G. (eds) Public Key Cryptography — PKC 2003. PKC 2003. Lecture Notes in Computer Science, vol 2567. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36288-6_2

[3] Bluetooth SIG. 2025. Bluetooth Core Specification v6.1.

[4] Abhishek, Kunal. (2023). Capability Review of X.509v3 with Link Maximum Transmission Unit in Public Key Infrastructure. 10.21203/rs.3.rs-3534862/v1.

[5] M. Troncoso and B. Hale. The bluetooth CYBORG: Analysis of the full human machine passkey entry AKE protocol. In NDSS 2021, 2021.

[6] M.vonTschirschnitz,L.Peuckert,F.Franzen,andJ.Grossklags. Method confusion attack on bluetooth pairing. In 2021 IEEE Symposium on Security and Privacy, pages 1332–1347, 2021.

[7] T. Claverie, G. Avoine, S. Delaune, and J. Lopes-Esteves. Tamarin-based analysis of bluetooth uncovers two practical pairing confusion attacks. In ESORICS 2023, Part III, pages 100–119, 2023.

[8] M. K. Jangid, Y. Zhang, and Z. Lin. Extrapolating formal analysis to uncover attacks in bluetooth passkey entry pairing. NDSS 2023, 2023.

[9] K. Haataja and P. Toivanen. Two practical man-in-the-middle attacks on bluetooth secure simple pairing and countermeasures. IEEE Transactions on Wireless Communications, 9(1):384–392, 2010.