S3-040481

Proposal for Enhancing Bluetooth Security Using an Improved Pairing Mechanism

Selim Aissi — Intel Corporation

Christian Gehrmann — Ericsson Mobile Platforms AB

Kaisa Nyberg — Nokia Research

Presented to the Bluetooth Architecture Review Board (BARB) at the Bluetooth All-Hands Meeting
April 19-23, 2004

Agenda

- Motivation
- The Bad News
- Introduction to Proposal
- Properties and Advantages of New Pairing Scheme
- Problems in Current Pairing Procedure: Some Technical Reports
- Pairing with a Network Access Point and a with a Limited Device
- Required Baseband and LMP Changes
- Security of the Protocol: Analysis
- Implementation Complexity of the Protocol: Analysis

Motivation

- **Pairing Process is Confusing to Users**
  - Having to enter the same passkey on both devices is unfamiliar
  - Current pairing mechanism is very well suited for "on the fly" or "ad hoc" generation of a shared secret between Bluetooth devices
    - When short passkeys are used, the pairing mechanism is vulnerable to both **on-line** and **off-line attacks**
    - This causes problems for Bluetooth applications with **high security** requirements
  - Entering a long passkey value is often not considered to be an acceptable solution, since it is **not very user-friendly**
  - Bluetooth SIM Access Profile (SAP) **requires** passkey values of 16 decimal digits
    - Since security is of utmost importance in this profile, **long passkeys** were considered as the **only** acceptable option despite their usability problems
- **Pairing is Currently Insecure**
  - Users have tendency to choose **short**, **simple** passkeys (“**human-friendly**” PINs)
  - Some devices have **limited** input capabilities, so passkeys are made up of a limited set of characters (i.e. phones with only digits).
  - Some devices don’t have any input capabilities, so must have a **fixed** passkey
- **Current and Emerging Market Demand for More Secure Bluetooth**
  - 3GPP WLAN/WWAN Interworking requirements (2LSs, etc.)
    - Eric Gauthier (Orange) WWAN/WLAN MiM Attack
    - WWAN credentials also exposed!
  - 3GPP SIM-Reuse requirements
  - Notebook-Handheld Interaction

Bluetooth

The Bad News

Bluestumbler

11/13/03

Serious flaws in the authentication and/or data transfer mechanisms on Bluetooth enabled devices.

BBC NEWS

Pickpockets turn to technology

Mark Ward, BBC News Online technology correspondent
11/17/2003

A potential loophole in security for Bluetooth phones, which could see strangers hacking into your address books, has been uncovered.

The Register

Bluetooth is attack vector for mobile phones

FEATURE STORY - Insights into the Latest Wireless LAN Security Issues
WLANs at CTIA Fall Short of Secure

http://www.airdefense.net/eNewsletters/Mar04/feature.shtm (3/23/04)

AirDefense found the following operating infrastructure: 216 access points, 24 Soft APs, laptops that function as access points, 609 user stations, **969 Bluetooth devices**, 42 ad-hoc networks

AirDefense research found numerous risks and threats including:

- 25 identity theft attacks on T-Mobile/Cisco sponsored Hotspot - intruders stealing identity of unsecured users to connect to the network without being charged
- **48 BlueSnarf**, a tool to connect to an unsecured device to gain access to restricted portions of data
- **393 BlueJack attacks** including the sending of "MyDOOM," "Your Cute" and "You Have WON" viruses

Introduction to Proposal

Bluetooth Pairing

- Utilizes an **initial secret** (passkey or PIN)
- At the pairing occasion, the **user** is asked to **enter** a PIN into the devices
- The PIN is then used to **calculate** an **initialization key**
- Using the initialization key, a common **link key** is derived
- This shared link key is used to **protect** subsequent communication between devices

Security Weaknesses

- Current pairing mechanism has been **criticized** on several occasions and in several research papers during the last two years
- Human users tend to use rather **short passkeys** (around 4 digits)
  - pairing mechanism is sensitive to **Passive eavesdropping** or a **Man-in-the-Middle (MiM) attack**
- As long as the passkeys are less than **16 decimal digits**, these attacks allow the attacker to obtain the secret **link key** with rather **small** computational effort

Improved Pairing Proposal

- This problem has encouraged the SEG to suggest an **improved** pairing scheme
- The new protocol makes use of **cryptographic one-way functions** to strengthen the security of the link key allowing at the same time a **user-friendly** short PIN

Properties and Advantages of New Pairing Scheme

One-way cryptographic functions

- New pairing protocol uses **Diffie-Hellman (DH)** key exchange to improve the security of pairing
  - Makes passive wiretapping and off-line attacks **infeasible**
  - Offers **strong protection** against active MiM attacks using relatively **short** and user-friendly PIN values
- Efficient implementation of the DH protocol can be based on **standard Elliptic Curve (EC)** algebra

Short PIN values

- Length of PIN to be handled by users varies between 5-12 decimal digits
  - New interfaces will become available for communicating initial secrets (e.g., RFID, acoustic/imaging channels)
  - New channels may improve usability and allow exchange of longer PIN values at the same time
- However, the basic channel (PIN communicated by the user) is expected to **remain** as an option
- The new pairing protocol improves **usability** also in the case of these **manual** PIN exchange methods

Two-stage approach

- New protocol can be divided into **two stages**: Registration Stage and Key Establishment stage
  - Registration stage comprises of **initial key generation** by one of the devices and **exchange of identities and cryptographic verification values**
    - Cryptographic verification value (PIN) can be **entered and stored** in the devices after the first stage

=> PINs are readily **available** at the second stage, where the link key is established

- Second stage comprises of the Diffie-Hellman key exchange
- Division into stages can be **useful** when pairing with a **limited device** or a **network access point**

Properties & Advantages of New Pairing Scheme – Cont'd

Pairing with Network Access Points or Limited Devices

- New protocol also allows pairing with a Bluetooth device with **no online input interface**
  - Typical examples of such devices are network access points and very small devices
- Unlike current Bluetooth pairing, new protocol is **asymmetric** with respect to devices
- In network access, the accessing device is typically **authenticated** using some higher layer authentication mechanism
  - **=>** It is often **not** necessary to identify the accessing device in the initial Bluetooth pairing.
  - It may be desirable to authenticate the access point of the serving network, and establish an encrypted Bluetooth link from the accessing device to the access point
- Higher layer authentication protocol can be run **protected over** encrypted Bluetooth link

Personal Transaction Protocol (PTP)

- Final draft of the PTP [3] was released from Mobile Electronic Transaction (MET) forum
  - PTP used for exchange of security related information (e.g., authentication, signing of messages) between a Personal Trusted Device (PTD) and a second device
- When used with Bluetooth, PTP also requires **long passkey values**
- PTP message integrity protection key is also **derived** from the Bluetooth **link key**
  - ⇒ Security of the **link key generation** is **crucial** for the **total** security level of PTP

Short Range Financial Transaction (SRFT)

- Security requirements for different SRFT applications are also **very high**
  - **=>** It is anticipated that SRFT expert group [4] is **not** going to recommend the current Bluetooth pairing mechanism for such applications
  - ⇒ They do **not** accept that one should expect the user to enter long passkey values into his/her device
  - ⇒ Such a requirement **cannot** be satisfied using the current pairing mechanism.
- Current pairing used with **short** passkeys is considered **not secure enough** for security-critical applications

Problems in Current Pairing Procedure: Some Technical Reports

Security problem with short PIN values has been reported in several papers and official reports

- ***Jakobsson and Wetzel [5]:*** it would be **possible** to obtain the **link key** at the initialisation through passive eavesdropping or a man-in-the-middle attack
- ***NIST Report [6]:*** the problem with **short PIN** values is as one of the **main** Bluetooth security vulnerabilities (together with unit key usage and privacy attacks)
- ***Vainio [7]:*** discusses the **short PIN** code problem in his paper on Bluetooth security
- ***Kügler [8]:*** passive eavesdropping attack on the pairing is described. To circumvent the attack, authors suggest to use **long** PIN values

Pairing with a Network Access Point and a with a Limited Device

Pairing with a Network Access Point

- When pairing with network, BD1 represents the **network**
  - Two different network entities play the role of BD1
- **At stage A**, the role of BD1 is taken by a registration desk or similar service point, which the user of BD2 contacts for receiving the secret values (K and C)
- The memory of BD1 is typically realized as network database, where the details of BD2 are stored
  - This database **must** be accessible online to all access points on the network
- **At stage B,** the network entity represented by BD1 is a network access point
- The access point (BD1) must first ask for a (temporary) identity of BD2 under which the specific public key values of BD2 are stored in the database.
  - It is also possible to use the same public key values for a number of BD2 devices and in this manner **avoid** asking the identity of BD2 at this step

Pairing with a Limited Device

- A simple device with **no human operable online interface** takes the role of the first device BD1
- Such device is assumed to have a management interface that can be operated **offline** by a key generation facility
- The keys and authentication values are transferred to BD1 using the management interface
  - The key management facility also outputs C and the key K on a piece of paper
  - The paper is delivered to the user of BD1 to be used when the pairing with BD2 is performed
- The authentication values are **not** revealed in the authentication exchange

  => It is possible to use them **more than once**
- This property of the new pairing protocol means **significant improvement** compared to current pairing method (where constant PIN is not secure)

Required Baseband and LMP Changes

Required Baseband Changes

- In order to support the improved pairing proposal, the following changes to the Baseband Specification are needed:
  - New pairing protocol must be described in detail
  - DH parameters and key generation must be specified
  - KDF function for derivation of the link key from the DH shared secret must be specified
  - MAC code (and hash function) must be specified
  - MAC key encryption must be defined
  - Possible new key IDs used in the pairing must be specified
  - If PIN is expressed in decimal digits, an efficient conversion (binary to decimal) must be specified.

Required LMP Changes

- Need a set of new LMP commands to support new pairing protocol
- Also feature mask needs to be updated to indicate the support of the improved pairing protocol
- Only a **small** set of new LMP PDUs are needed
- DH public values might be quite large (160 bits or 20 bytes)
  - They need to be **fragmented** like the existing *LMP_name_req PDU*

Note: LMP=Link Manager Protocol

Security of the Protocol: Analysis

- Security of protocol **depends** on: 1. length of PIN, 2. security of DH protocol, 3. hash function, 4. MAC function
- The DH key exchange and the one-way hash function are **computationally secure**
  - **=>** The **risk** of **off-line attacks** on the pairing is **eliminated**

=> Security of the scheme depends **purely** on PIN length (formed by authentication key and MAC code)

- The most powerful attack that **remains** is an **active MiM attack**
  - **=>** Given a DH public key, the attacker **must** find another DH key with the same C without knowledge of K and C
  - **=>** Table below lists the **probabilities** for successful man-in-the-middle attacks on the improved pairing protocol
- Probabilities vary within given interval depending on how much off-line analysis the attacker is capable of

Note: it is assumed that the MAC code is based on a **Reed-Solomon** construction

Size of message hash (bits)	PIN size (decimal digits)	PIN size (bytes)	Probability	Probability in current BT system
128	5	2	2^{-4} - 2^{-8}	1
128	8	3	2^{-8} - 2^{-12}	1
128	10	4	2^{-13} - 2^{-16}	1
256	10	4	2^{-12} - 2^{-16}	1
128	12	5	2^{-17} - 2^{-20}	1
256	12	5	2^{-16} - 2^{-20}	1

1. **Probability rapidly decreases with increased PIN size for the chosen MAC.**
2. **For small PIN sizes like 2 bytes (5 decimal digits), probability of a successful man-in-the-middle is at most 2^{-4}.**
3. **For most applications, PIN size of 4 bytes (10 decimal digits) provides reasonable security level.**

Implementation Complexity of the Protocol: Analysis

- Using the new improved pairing based on **ordinary** DH would cause considerable increase in bandwidth and implementation costs compared to current Bluetooth modules

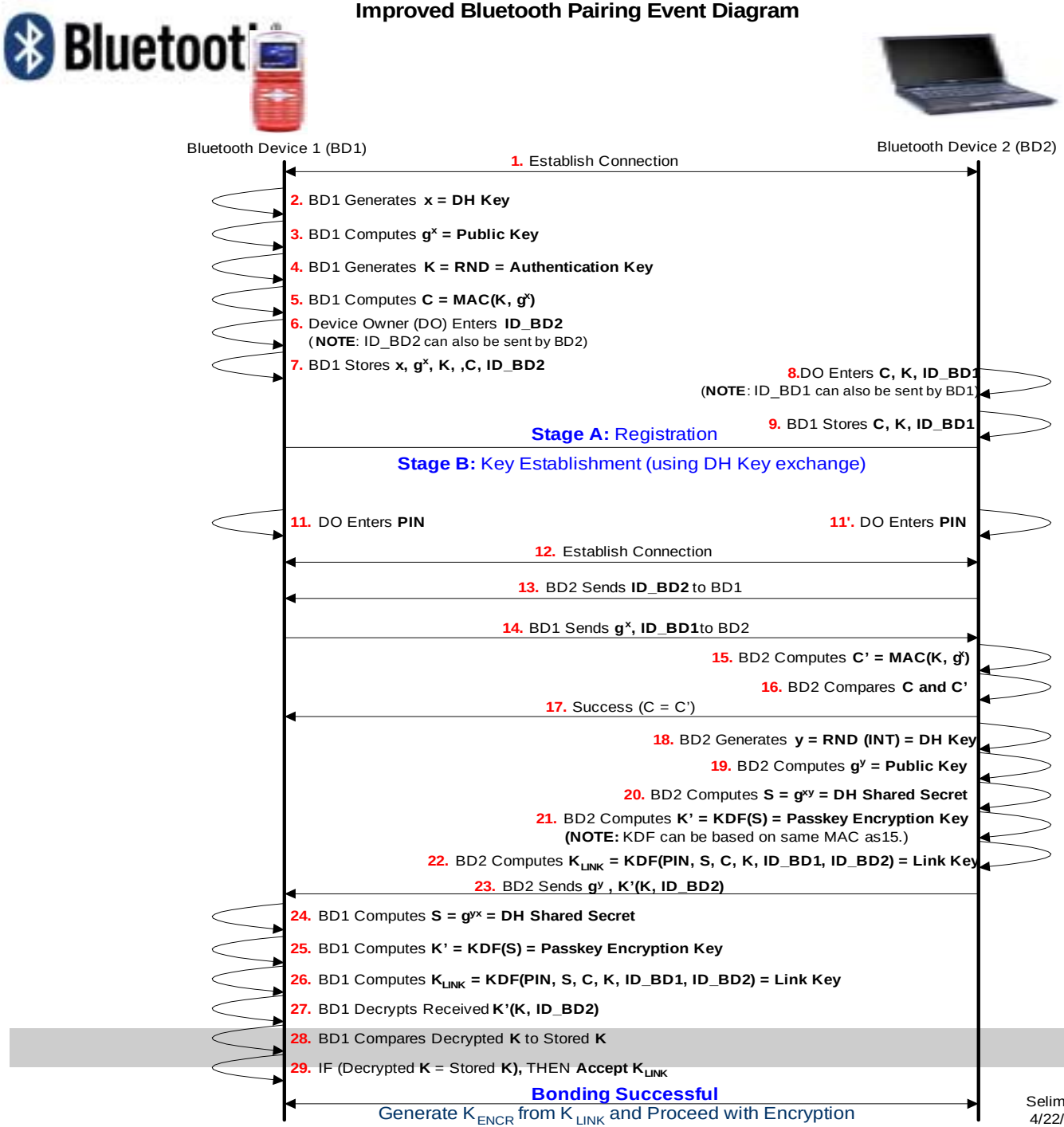
  => Suggest that **Elliptic Curve Diffie-Hellman** (**ECDH**) [13] is used instead

- Using ECDH
  - We estimate the additional hardware requirements to **around 10k gates** for a very fast computation or less than 10kB footprint for a pure software implementation [1]
  - For an elliptic curve over a field of size around 2160, i.e., 160 bits size of the underlying field and with a clock speed at 10MHz, we will be able to calculate the secret key with a hardware accelerator solution in **less than 0.1 second**

  => We have public key size also of about **160 bits** (20 bytes)

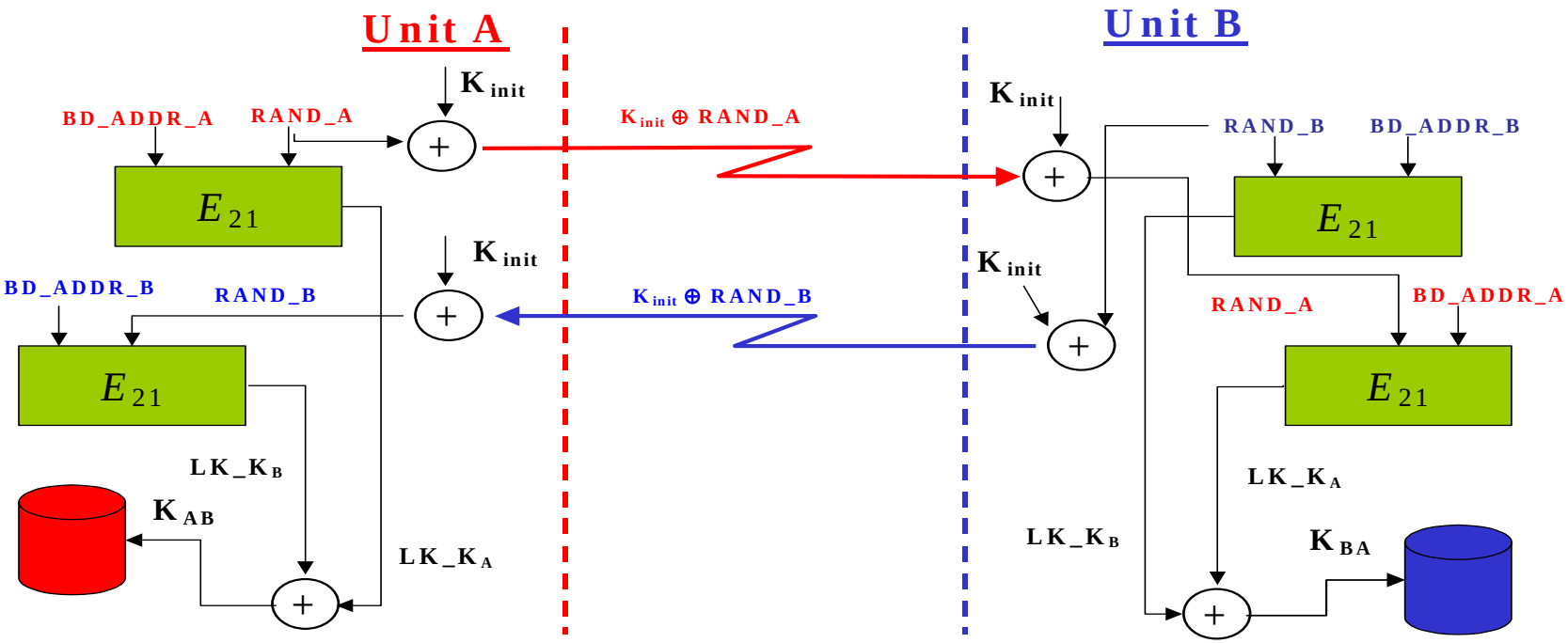
  - A key at this size needs then to be sent over the radio channel at the pairing
- The **choice** of elliptic curve parameters is for **further evaluation**
  - We should use **standard curves**: IEEE 1363 [14], ANSI X9.63 [15] or NIST [16]

Improved Bluetooth Pairing Event Diagram



Backup

Current Pairing Procedure



- The current Bluetooth **Combination Key** is calculated as shown.
- K_{init} = **Initialization Key** .
- K_{init} is derived as the output of an algorithm called E22 prior the generation of the **Combination Key** .
- E22 takes as input the address of one of the Bluetooth units, BD_ADDR_A, RAND, and the PIN, i.e., K_{init}_A = E22(BD_ADDR_A, RAND, PIN).
- RAND is sent in **clear text** between the two units over the Bluetooth radio channel.
- As shown in above, the **Initialization Key** is then used to encrypt random values, RAND_A and RAND_B, which are used to derive the **Combination Key** , K_{AB} (a similar procedure is used to exchange a unit key).

Problems in Current Pairing Procedure

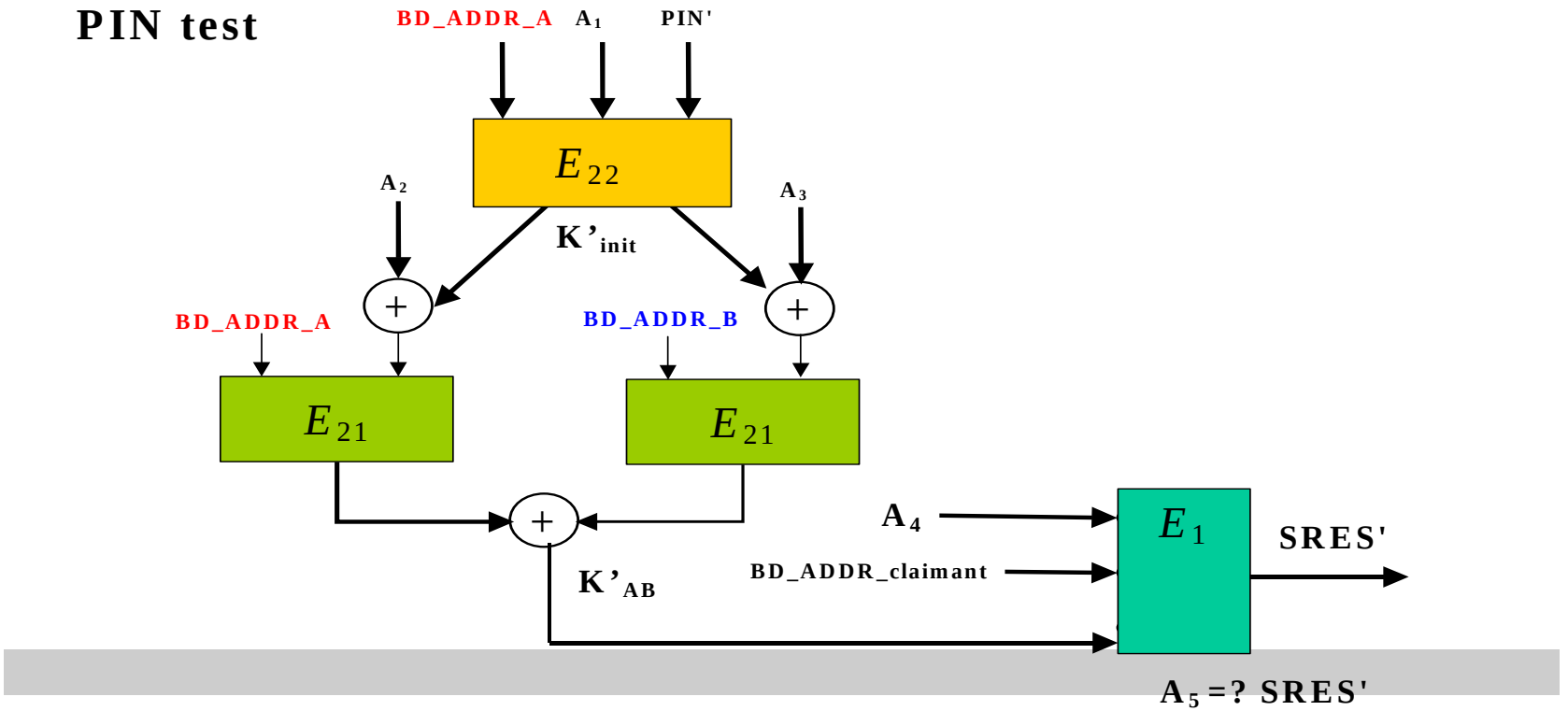
- A third party who observes all the communications between A and B during the pairing procedure obtains the device addresses BD_ADDR_A and BD_ADDR_B, the random value RAND, and the encrypted random values $K_{init} \oplus$RAND_A and $K_{init} \oplus$RAND_B.
- Hence, the **only** unknown parameter used in the calculations of K_{AB} is the PIN.
- Given that an attacker observes all these values, he might then try to guess which PIN value was used during the pairing and the corresponding link key value.
- However, in order to check if the guess is correct, the attacker must have some additional information.
- This information is obtained by observing the authentication message exchange that always follows the link key calculation exchanges.
- At the authenticating procedure the verifier sends a random value, AU_RAND to the claimant unit.
- The claimant then sends a response, SRES = E2(BD_ADDR_claimant, AU_RAND, KAB), where E2 is the Bluetooth authentication algorithm.
- Hence, the attacker can **observe** the following parameters during the pairing procedure:
  - A1 = RAND
  - A2 = $K_{init} \oplus$ RAND_A
  - A3 = $K_{init} \oplus$ RAND_B
  - A4 = AU_RAND
  - A5 = SRES

Problems in Current Pairing Procedure: PIN Test

- Using these observations, the attacker can **successively** create some passkey values PIN' and calculate the corresponding link key, K'AB.
- Given the observed values A1,... A4 and each guess of the passkey value, corresponding SRES' **can then be calculated**.
- If the calculated value equals the observed value SRES, the attacker might have **guessed correctly**.
- If the size of PIN < SRES (32 bits or about 9 decimal digits), he can be almost sure that the guess was **correct**.
- If further confidence is needed, the second authentication exchange can be used to verify the guess (mutual authentication is always performed at the pairing).

  => *If the PIN is **short,** attacker **can** check all possible values **and see** where he gets **SRES' = SRES.***

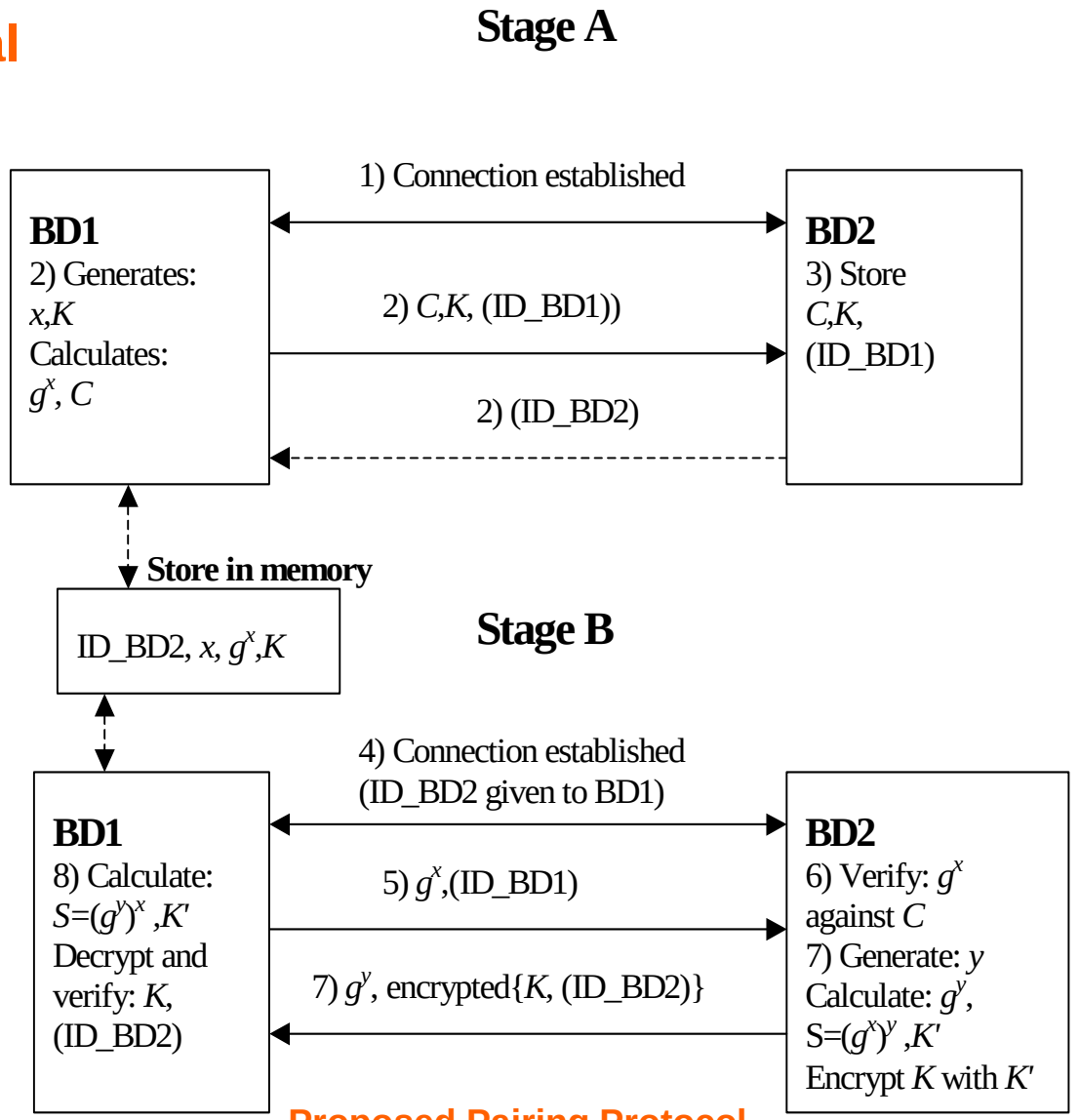
  => *Short PIN values do **not** protect users from a **passive eavesdropper** or **man-in-the-middle** present during pairing.*



Improved Pairing Proposal

The Basic Protocol

- The steps of the protocol are separated into **two** stages, stage A and B.
- Stage A is performed before stage B, but how long before will depend on the usage scenario.
- When two peer devices are paired, typically no break between stages A and B is necessary.
- However, when pairing takes place with a network access point, or with a device with no interface that can be securely operated online, then stages A an B are separated.

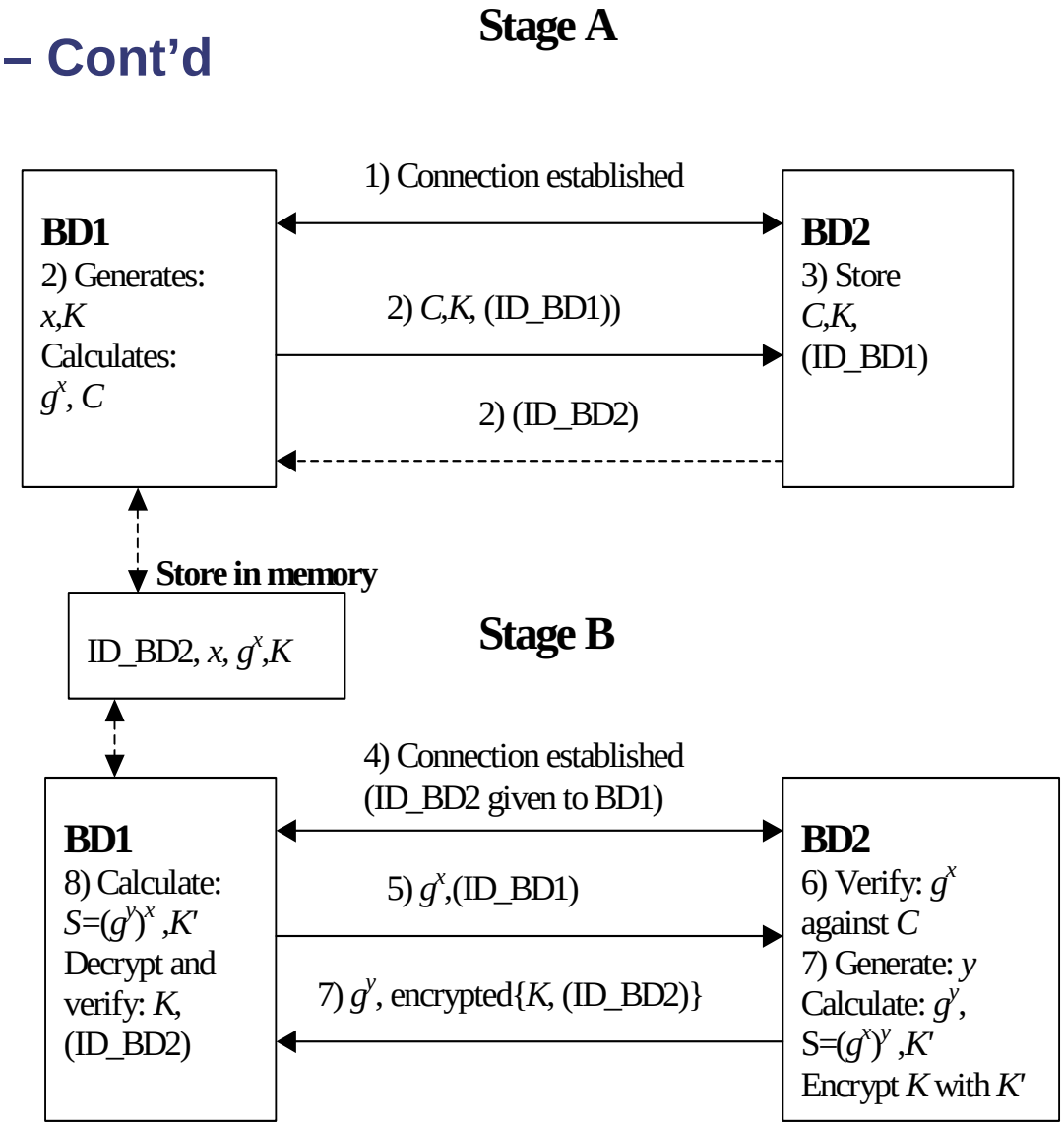


Proposed Pairing Protocol

Improved Pairing Proposal – Cont'd

The Basic Protocol

- In this protocol it is assumed that BD1 has an output interface using a short secret PIN value which can be displayed secretly to the user, who then will enter it to BD2.
- The PIN can also be transported from BD1 directly to BD2 using some other suitable and secure physical transport channel.
- BD2 is assumed to have an appropriate input interface.
- Both devices perform the same cryptographic computations.
- The main difference between BD1 and BD2 is in which interface they use and how they handle PIN values.
- This also changes the order of computations.
  - Typically, a Bluetooth device can act in the role of BD1 or BD2.

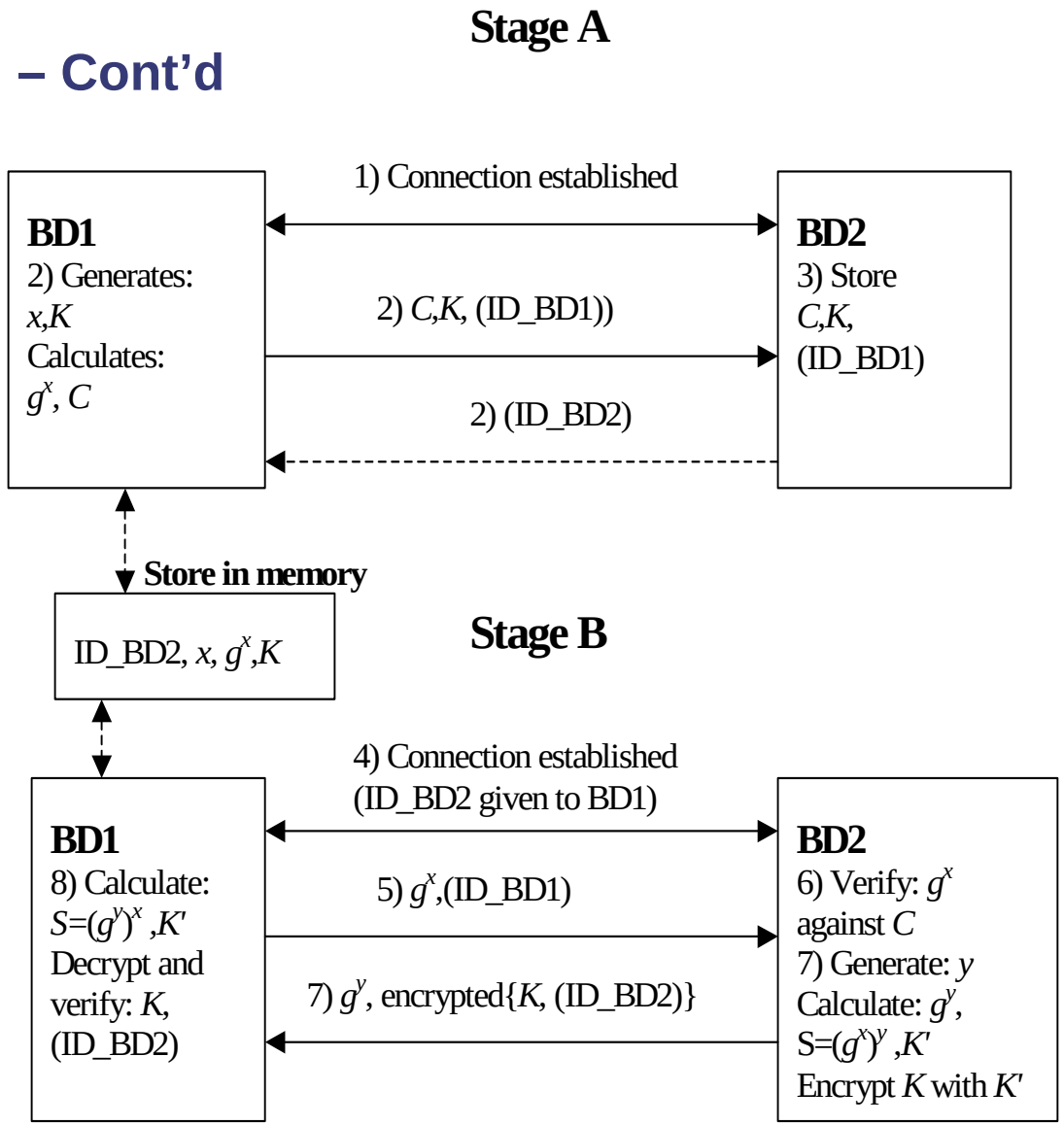


Proposed Pairing Protocol

Improved Pairing Proposal – Cont'd

The Basic Protocol

- Before the pairing can start, roles are **negotiated**.
- Network access points and limited devices with no online operated user interface **always** act in the role of BD1.
- It is assumed that the parties **share a function** for computing message authentication codes (MAC function) and a **key derivation function** (KDF).
  - Later, a recommended MAC function is described (makes use of a cryptographic hash-function and unconditionally secure MAC).
  - Key derivation function can be based on same hash function.
- Proposed protocol also makes use of an **encryption mechanism**, which can be a simple XOR-based encryption.



Proposed Pairing Protocol

Improved Pairing Proposal – Cont'd

Stage A

1. Stage A is **initialized**.

2. BD1 generates its **secret DH key x** and computes the **public key g^x**. BD1 also generates a **short random key K**, where K is suitable for use with the MAC function shared by the two devices.

   -- Then it calculates the C = MAC(K, g^x) with key K on the data comprising of g^x and possibly some other data. The BD1 stores the x, g^x, and K in its memory .

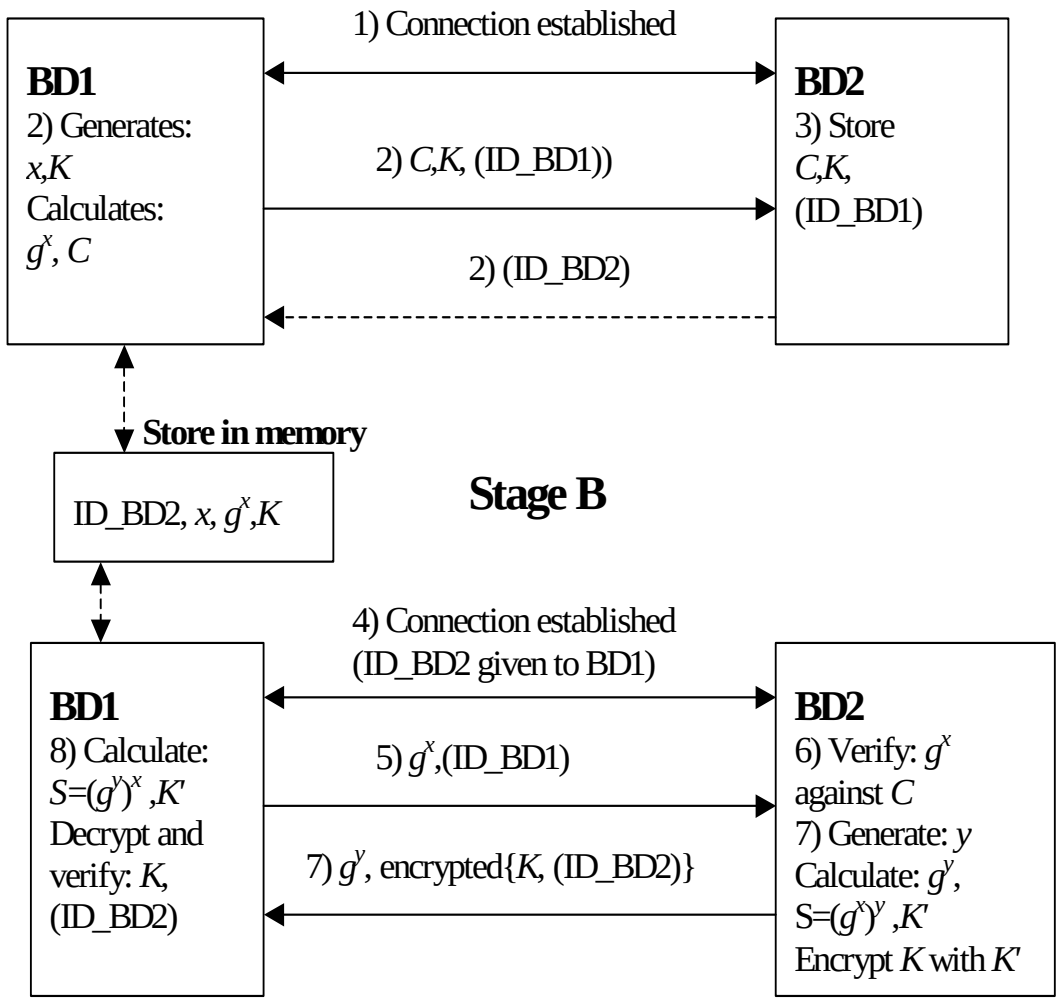
   -- In cases where more than one pairing may be taking place concurrently, BD1 also asks for the (temporary) identity of BD2 and stores this value together with the keys in its database.

   -- The MAC and the key K are then given to the user of BD2 for example through the output interface of BD1.

3. User now reads the code C and the key K possibly together with the identity of BD1 and enters them to BD2 using the human operable online interface.

   -- The K and C values are stored in BD2. –

   -- The K and C values can also be entered and stored in some user readable medium to be entered to BD2 later in the beginning of Stage B (so that it is available when needed in step 6).

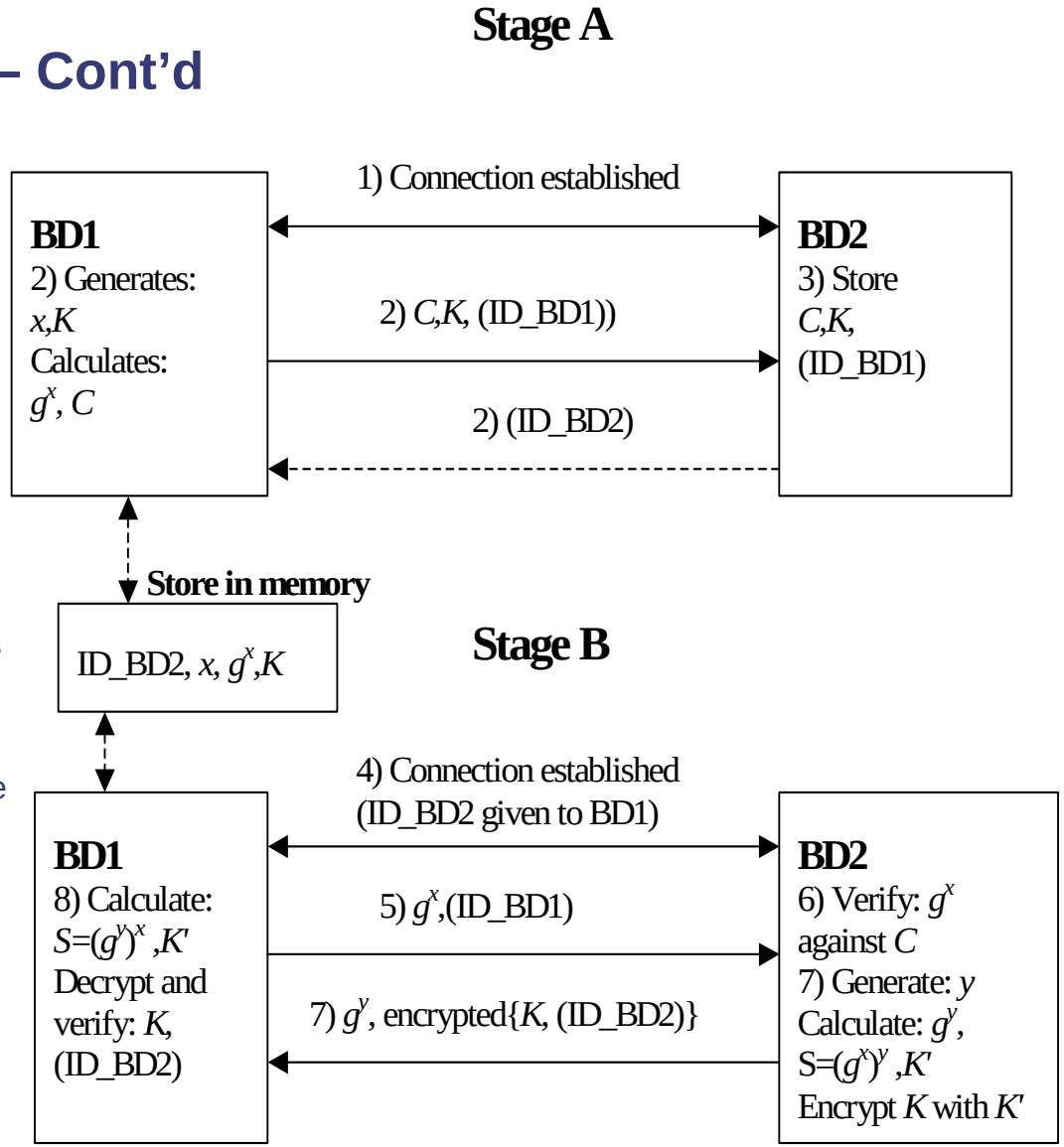


Proposed Pairing Protocol

Improved Pairing Proposal – Cont'd

Stage B

4. Stage B is **initialized**.
5. BD1 sends g^x and possibly some other data like its identity to BD2.
6. BD2 receives g^x. It verifies the authenticity of g^x and possibly of other received data using the stored values of K and C as follows. BD2 uses the key K to re-compute the C value as a function of the received data. If the two C values agree then BD2 accepts the data and outputs a success signal to the user. Otherwise, failure signal.
7. BD2 generates randomly and privately an integer y, and computes g^y. Then it computes the Diffie-Hellman shared secret as $S = (g^x)^y$ and uses S as input to the KDF to derive the passkey encryption key K' and the link key. Then it sends its public key g^y in clear and the key K (together with its identity) encrypted using the key K'.
8. BD1 computes its copy of the shared secret as $S = (g^y)^x$. Further, it uses S as input to the KDF to derive its copy of the link key and the encryption key K'. Then it decrypts the encrypted K and the identity of BD2. If the decrypted value for K is equal to K stored by BD1 for this identity, then BD1 accepts the link key.



Proposed Pairing Protocol

MAC Constructions and Diffie-Hellman Values

- In order to achieve **high security** with short K and C values, the MAC codes used in the enhanced protocol must be constructed in a certain way
  - The best choice is to use a **secure one-way hash function like SHA-1** (or a hash function based on the existing SAFER+) and an unconditionally secure message authentication code.
  - Such code can be constructed in a practical way from codes with large minimum distance such as **Reed-Solomon codes** [12, 11]
- Further optimizations of the constructions can be achieved [10]
  - Suggest to use the **Reed-Solomon based MAC constructions** in the enhanced pairing protocols
- The same user operated PIN value can be used **more than once**, but preferably only a small limited number of times
- The PIN must be kept **secret** as long as it is going to be used for pairing
  - Any party in possession of a valid key K and a MAC code C would be able to impersonate the first device (requires some extra effort) or the second device (easily)
- If the PIN is revealed, **no danger** is caused to previous pairings where it was used
- This is a **significant improvement** compared to the current pairing procedure

MAC Constructions and Diffie-Hellman Values – Cont'd

- **One DH** public key can remain **constant**.
  - This feature can be exploited in the network access case to make it simpler to handle different users.
- If considered important, it would also be possible to allow **both DH** public keys remain **constant**.
  - The parties must exchange some fresh random values to take as input to the key derivation function so that a fresh link key is produced at each new pairing.
- If **both** DH keys are constant,

  => It is possible to do pairing if only one of the devices has a human operable input interface
- Using constant DH keys does **not** change the fact that every device must be capable of computing the DH shared secret using its secret key, that is, at least one exponentiation in a large finite group.
- The Diffie-Hellman parameters can be chosen in many different ways.
  - To provide **small footprint implementation** and as small key exchange values, **ECC-based DH** key exchange is **recommended**

References

[1] G.B.Agnew, R.C.Mullin and S.A.Vanstone, “An Implementation of Elliptic Curve Cryptosystems Over F2155, IEEE Journal on selected areas in communications, Vol 11, No 5, June 1993.

[2] Bluetooth SIG, "SIM Access profile", Version 0.95, December 2002.

[3] MET, "Personal Transaction Protocol Version 1.0", draft, November 2002, http://www.mobiletransaction.org/

[4] Bluetooth SIG, " Short Range Financial Transactions Using Bluetooth white paper", version 0.1.11.3, December 2002.

[5] M. Jakobsson, S. Wetzel: “Security Weaknesses in Bluetooth”, Proceedings of the RSA Conference 2001, San Francisco, USA, April 8 – 12, 2001, Springer Lecture Notes in Computer Science Vol. 2020, ISBN 3-540-41898-9. http://www.rsasecurity.com/rsalabs/staff/bios/mjakobsson/bluetooth/bluetooth.pdf

[6] T. Karygiannis and L. Owens, " Wireless Network Security, 802.11, Bluetooth and handheld devices", NIST special publication 800-48, November 2002

[7] Vainio, Juha, “Bluetooth Security”, May 2000, at http://www.niksula.cs.hut.fi/~jiitv/bluesec.html

[8] D. Kügler, "Preventing Tracking and "Man in the Middle" Attacks on Bluetooth Devices", Proceedings of Financial Cryptography 03.

[9] Shaman, Deliverable D07, http://www.ist-shaman.org/

[10] Shaman, Deliverable D13 http://www.ist-shaman.org/

[11] G. Kabatianskii, B. Smeets and T. Johansson, “On the cardinality of systematic A-codes via error correcting codes”, IEEE Transaction on Information Theory, vol. IT-42, pp. 566-578, 1996.

[12] I. S. Reed and G. Solomon, “Polynomial Codes over Certain Finite Fields”, Journal of the Society for Industrial and Applied Mathematics. Vol. 8, pp. 300-304, 1960.

[13] A. J. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer, 1993.

[14] IEEE, "Standard Specifications for Public Key Cryptography", IEEE Std 1353-2000, 2000

[15] ANSI, "Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport using Elliptic Curve Cryptography", ANSI X.9.63, 2001

[16] NIST, "Secure Hash Standard", FIPS 186-2, 1995.