Evaluation of anonymous digital signatures for privacy-enhancing mobile applications

Lukas Malina*, Petr Dzurenda and Jan Hajny

Department of Telecommunications, Brno University of Technology, Brno, Czech Republic
Email: malina@feec.vutbr.cz
Email: dzurenda@feec.vutbr.cz
Email: hajny@feec.vutbr.cz
*Corresponding author

Abstract: Privacy-enhancing mobile applications that employ anonymous digital signature (ADS) schemes can be beneficial for users who strongly concern about their privacy and security. This paper deals with ADS schemes and their usability on smartphones. We implement seven ADS schemes on smartphones in order to get the practical results. Further, we discuss the usability of these schemes in different scenarios such as data collection, access control and data notification. We believe that our analysis and performance assessment can help engineers and security experts to choose a proper ADS scheme into their secure and privacy-friendly mobile applications.

Keywords: ADS; anonymous digital signature; cryptography; group signature; performance evaluation; privacy.


Biographical notes: Lukas Malina is a Researcher at the Department of Telecommunications of the Faculty of Electrical Engineering and Communication (FEEC) at Brno University of Technology (BUT), Czech Republic. He accomplished his MSc degree with honors and the Dean prize at BUT in 2010. He received his PhD from BUT in 2014. During his internship at Universitat Rovira i Virgili, Tarragona, Spain 2011–2012, he designed new group signature schemes. Currently, he deals with the privacy-preserving cryptographic protocols and authentication systems on constrained devices. He has published more than 50 papers in international journals and conferences, and he is involved as a researcher in several Czech scientific projects. He is a Member of the Advanced Cybersecurity Research Group at BUT.

Petr Dzurenda is a Junior Researcher at the Department of Telecommunications at Brno University of Technology (BUT), Czech Republic. He accomplished his MSc degree at BUT in 2013. He deals with authentication schemes and developing authentication systems and secure communication protocols on smartcards and smartphones. Further, he is interested in applied cryptography, computer security and network security. He has published more than 15 papers in international journals and international and national conferences. He is involved as a developer in several Czech scientific projects.

Jan Hajny is an Associated Professor at the Department of Telecommunications at BUT. Besides internship at the University of Minnesota, USA, from 2010 to 2011, he studied at the Department of Computer Science (Crypto-Group), the University of Arhus, Denmark in 2008. As a Senior Researcher, he deals with the design of privacy-enhancing cryptographic protocols and authentication schemes. He is involved as a team leader in several industry-oriented projects, basic research projects, and he leads the Advanced Cybersecurity Research Group at BUT.

1 Introduction

Mobile applications and services that are required to protect user privacy and keep security properties have to often employ modern cryptographic schemes, e.g., group signature schemes, attribute-based signature schemes, anonymous credential schemes. These schemes are designed to ensure security and privacy properties such as non-repudiation, data
integrity, data authenticity, anonymity, user revocation etc. These privacy-preserving signature schemes are sometimes called anonymous digital signatures (ADS). The standard ISO/IEC 20008-2:2013 (ISO/IEC 20008-2, 2013) provides the general description of ADS mechanisms that use a group public key. A signer is able to generate a signature on behalf of the group and a verifier is able to verify a signed message by the group public key. These schemes are usually based on zero knowledge and the proof of knowledge protocols and provide advanced security properties such as soundness, completeness, anonymity, unforgeability, traceability, coalition resistance, non-frameability and unlinkability (more detailed definitions can be found in Chaum and Van Heyst (1991) and Bellare et al. (2003)).

ADS can be beneficial in many smartphone applications where the privacy protection of users is required. Nonetheless, these schemes are more computationally expensive and can produce larger signatures than conventional digital signature schemes such as RSA or ECDSA. In the following subsections, we present three scenarios with ADS schemes that are using smartphones and other smart devices (e.g., tablets, smartwatches). These scenarios represent data collection, access control and data notification systems that have different communication, security and practical requirements.

1.1 Scenario 1: Privacy-preserving data collection

Scenario 1 (depicted in Figure 1) represents a general data collection system which uses smartphones, tablets and other handheld devices. Users create signatures on their messages that are sent to a server managed by a service provider. If some unsecured services work with sensitive and personal user data, then there is the risk that the data may leak to some third parties without user’s permission. Furthermore, the attackers can tamper with messages and users can deny their creation without the deployment of digital signature or digest mechanisms. However, anonymous signature schemes should provide security to the provider and privacy to users in this scenario. Thus, users’ messages are signed, protected against tampering and an unauthorised party is not able to link the messages from the same user. Privacy-preserving data collection can be used for example in automated smartphone data collection systems (Mccann, 2016; Onishi and Asaka, 2016) and in health-system data collection (Sa et al., 2016). The data can be collected in short-time periods (e.g., less than 150 ms, online data collection) or long-time periods (once per time, offline data collection). Users with their smartphones send data to a server which collects data and processes them. The anonymous data collection based on group signatures is also used in a real-use scenario covering data acquisition (e.g., signal strength) from smartphones (Troják and Komosný, 2016).

1.2 Scenario 2: Privacy-preserving access control

ADS can serve in access control systems and applications that ensure the privacy of users. Access control systems can protect physical or digital assets such as room/building entrances, online services, subscription memberships, data storages and clouds. Scenario 2 is depicted in Figure 2. Users can use their smartphones, smartwatches and other handheld devices instead of chip cards and RFID tags. If smartphones are able to communicate with a reader or a verifier then a user can authenticate themselves and lock/unlock protected services. In the authentication phase, users sign challenge messages obtained from the verifier by the ADS scheme. Smartphones are more powerful than chip cards and enable to employ more advanced cryptographic protections. Therefore, we expect that smartphones will be used more and more in these services in future. There are several works that deal with privacy-preserving access control, e.g., Camenisch and Van Herreweghen (2002), Li et al. (2008), Hajny and Malina (2013) and Hajny et al. (2015). For example, the IBM Identity Mixer protocol suite can provide privacy-preserving authentication and access control. This protocol allows user authentication without divulging any personal data.

Figure 1 Scenario 1: privacy-preserving data collection (see online version for colours)
1.3 Scenario 3: Privacy-preserving data notification

In Scenario 3 (depicted in Figure 3), user messages and notification data are broadcasted from users to other users. In this communication pattern, ADSs are usually used to secure data that are broadcasted or uploaded. This scenario offers anonymous data uploading, notification and sharing. Users with their smartphones or handheld devices directly broadcast data via an access point or a router or send data to servers that forward data to other users. Other users then check data signatures and accept messages if the signatures are valid. There are systems where the privacy and anonymity of users are highly required, for example, vehicular ad hoc networks (VANETs) solutions published in papers (Guo et al., 2007; Wasef and Shen, 2010; Malina et al., 2015), vehicle to everything (V2X) solutions described in Yang et al. (2017), geo-social applications presented in Patel and Palomar (2014), Malina and Hajny (2014) and other anonymous broadcast communication systems, e.g., Stajano and Anderson (2000).

1.4 Our contribution and paper organisation

In this work, we provide the performance assessment of ADS. We evaluate anonymous digital schemes on smartphones with the Android platform, and we analyse their suitability in data notification, access control and data collection scenarios. We believe that this work can help with the research and development of smartphone applications using the ADSs. The contribution of this work is threefold:
We present the evaluation of seven ADS schemes, namely, the BBS scheme (Boneh et al., 2004), the DP scheme (Delerablée and Pointcheval, 2006), the HLCCN scheme (Hwang et al., 2011), the ACJT scheme (Ateniese et al., 2000), the CG scheme (Camenisch and Groth, 2005), the IMSTY scheme (Isshiki et al., 2006), the HM scheme (Hajny and Malina, 2013). We show which parameters can be precomputed and cached to get higher performance. The properties of these seven schemes are compared also with two recent ADS schemes supporting linkability, namely, the HCCN scheme Hwang et al. (2015) and the EH scheme (Emura and Hayashi, 2015).

We implement seven ADS schemes on smartphones and present the practical results which show the runtime of main scheme phases such as signing and verification without and with optimisation tricks (e.g., full precomputation, pairing precomputation and collapsing).

We provide the interesting insights about which ADS schemes are better to use in the defined scenarios (online/offline data collection, access control and data notification).

The rest of this paper is organized as follows. Section 2 presents state of the art and works that deal with ADS implemented on smartphones and other handheld devices. Section 3 introduces chosen ADS schemes and their main phases. Section 4 presents the theoretical evaluation of the schemes. Section 5 shows our results of schemes from our experimental measurement. In Section 6, we discuss the perspective of anonymous digital schemes in privacy-enhancing smartphone services. Section 7 outlines our conclusions.

2 Related work

Anonymous digital signature schemes, which are analysed in Manulis et al. (2012), Arroyo et al. (2015), Martinez et al. (2015), are usually designed for more powerful devices (e.g., PCs and servers) than smartphones. Nevertheless, we focus mainly on the perspective and usability of these schemes on smartphones in this paper.

The most related work to this paper is the paper of Isern-Deyà et al. (2014) who present the performance analysis of two group signature schemes on mobile devices. They implement and test the pairing-based group signature scheme (i.e., the BBS scheme proposed by Boneh et al. (2004)) and the non-pairing group signature scheme (i.e., the ACJT scheme proposed by Ateniese et al. (2000)). Their results show that the signing and verification phases of both schemes take few seconds on the smartphones with the Android platform and 1-GHz CPU. The ACJT scheme takes 2.9 s for signing the message and 1.74 s for the verification of one message. The BBS scheme which uses the D pairing takes 16.53 s for signing one message and 25.17 s for verifying one signature. The BBS scheme with the D pairing and precomputation takes only 0.05 s for signing one message because all pairing operations can be precomputed. Nevertheless, the verification of one signature takes 14.14 s if the precomputation is used. The ACJT scheme with precomputation takes 0.01 s for signing one message and 1.4 s for the verification of one signature. In our paper, we evaluate, experimentally implement, and measure seven ADS schemes on smartphones with the Android platform. Besides BBS (Boneh et al., 2004) and ACJT (Ateniese et al., 2000) schemes as in the paper (Isern-Deyà et al., 2014), we add five different schemes: the DP scheme (Delerablée and Pointcheval, 2006), the HLCCN scheme (Hwang et al., 2011), the CG scheme (Camenisch and Groth, 2005), the Isshiki et al. (IMSTY) scheme (Isshiki et al., 2006) and the HM scheme (Hajny and Malina, 2013) to get better insights about the performance of ADSs.

Another closely related work is the paper written by Potzmader et al. (2013). The authors investigate the performance evaluation of ADS schemes on mobile devices. They present the framework written in JAVA which contains three group signature schemes, namely, the ECC-DAA scheme proposed by Chen et al. (2010), the Isshiki et al. scheme (Isshiki et al., 2006) and the Canard et al. scheme (Canard et al., 2006). These schemes are included in the upcoming ISO20008-2 standard (ISO/IEC 20008-2, 2013) which defines seven ADSs. They measure the signing (with/without precomputations) and verification phases of the group signature schemes on Android devices. Signing without precomputation by the Canard et al. scheme takes about 300 milliseconds on a 4 x 1.4-GHz ARM device. They argue that group signature creation with precomputation can be performed on mobile devices and its verification can be done on more powerful devices. In our paper, we significantly enhance this study by the practical comparison of seven ADS schemes. We also implement the Isshiki et al. scheme (in our paper designed as IMSTY) and we add six different ADS schemes into our evaluation.

There are more papers that deal with anonymous signature and group signature schemes and their usability on constrained devices such as smartphones, microcontrollers and so on. Spreitzer and Schmidt (2014) investigate group signature schemes, namely, the BBS scheme (Boneh et al., 2004), the Boneh and Shacham (BS) scheme (Boneh and Shacham, 2004), the Delerablée and Pointcheval (DP) scheme (Delerablée and Pointcheval, 2006), and the Hwang et al. (HLCCN) scheme (Hwang et al., 2011). They provide the comparison of these schemes including the number of the operations for signing and verification phases and signature sizes. They implement the BBS and HLCCN schemes on 32-MHz AVR microcontrollers by using an efficient Library for Cryptography called RELIC (Aranha and Gouvêa, 2013). Signing one message by the HLCCN scheme requires about 6 seconds. The BBS scheme with precomputed pairing operations is slightly faster than the HLCCN but it also takes about 6 s. Canard et al. (2011) implement the Delerablée and Pointcheval (DP) group signature scheme Delerablée and Pointcheval (2006) on sensor nodes (8-bit 7.37-MHz Atmega128 microprocessor and 16-bit 4-MHz MSP430) to prove that group signatures can be executed even on the restricted devices such as sensors. Nevertheless, they employ...
the cooperative sign procedure which uses an intermediary (I). I compute all the operations that do not need the group member signature key of the signer. Further, some expensive operations have to be precomputed or loaded as coupons. Finally, the online signing phase takes less than 200 milliseconds. On the other hand, the off-line signing phase needs about several seconds. The cooperative mode with an intermediary server uses the standard generalisation of Schnorr’s protocol (Schnorr, 1991) for proving the knowledge of the discrete logarithm. This group signature scheme consists of these phases: key generation, sign, verify and open.

The sign phase generates a signature σ on a message \( M \in [0,1]^* \) by using a member secret key \( gsk = (A,x) \) and a group public key \( gpk = (g_1,g_2,h,u,v,w = g_2^h) \), where γ is a secret issuer key. The signature is computed by the zero-knowledge protocol of the Strong Diffie-Hellman assumption. A signer randomly selects exponents \( a, b, \gamma \in \mathbb{Z}_p \) and computes the linear encryption of \( A \) represented by values \( T_1, T_2, T_3, \ldots \):

\[
T_1 = u^a, T_2 = v^b, T_3 = Ah^{a+b}, \\
\delta_1 = ax, \delta_2 = bx.
\]

The blinding values \( r_x, r_y, r_z, r_{\delta_1}, r_{\delta_2} \) are randomly picked from \( \mathbb{Z}_p \), and values \( R_1, R_2, R_3, R_4, R_5 \) are computed:

\[
R_1 = u^a, R_3 = e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_x-r_y} \cdot e(h, g_2)^{-r_y}, \\
R_2 = v^b, R_4 = T_1^a u^{-r_x} R_5 = T_3^b e^{-r_y}.
\]

The signer computes a challenge \( c \in \mathbb{Z}_p \) using the hash function \( H \) and values \( s_a, s_b, s_x, s_y, s_z, s_{\delta_1}, s_{\delta_2} \) to seal the proof of the knowledge of \( (a, b, x, \delta_1, \delta_2) \):

\[
c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), s_a = r_a + ca, \\
s_b = r_b + cb, s_x = r_x + cx, s_y = r_y + c\delta_1, s_z = r_z + c\delta_2.
\]

The signer outputs the signature \( \sigma = (T_1, T_2, T_3, c, s_a, s_b, s_x, s_y, s_z, s_{\delta_1}, s_{\delta_2}) \). All pairing operations can be precomputed and cached (pairing precomputation) because the input parameters are static. If the signer uses the full precomputation then all values are generated and the parameters in equations 1 and 1 are computed in advance. The signer computes only the parameters in equation (1).

In the verification phase, a verifier checks the validity of the signature \( \sigma \) generated on the message \( M \) by using the group public key \( gpk = (g_1, g_2, h, u, v, w) \). All the values \( R_1, R_2, R_3, R_4, R_5, c \) are restored:

\[
R'_1 = u^a T_1^{-c}, R'_2 = v^b T_2^{-c}, R'_4 = u^{-s_a} T_3^a, R'_5 = v^{-s_z} T_3^b T_2^c, \\
R'_3 = e(T_3, g_2)^{r_x} e(h, w)^{-r_x-r_y} e(h, g_2)^{r_y}, \\
\]

\[
e(h, g_2)^{-r_x-r_y} e(T_3, w) e(g_1, g_2) e^{-r_y}, \\
c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5).
\]

If \( c \) is equal to restored \( c' \), then the verifier accepts the signature and rejects otherwise. All pairing operations with static values \( e(h, w), e(h, g_2), e(g_1, g_2) \) can be precomputed in advance. The pairings \( e(T_3, g_2) \) and \( e(T_3, w) \) can be collapsed into one pairing operation.

### 3.1.2 Delerablee and Pointcheval (DP) scheme

The security of the DP scheme (Delerablee and Pointcheval, 2006) holds under the q-SDH and the XDH assumptions (also DLIN assumption can be used), in the random oracle model. The scheme improves the security (anonymity and non-frameability) of the BBS scheme (Boneh et al., 2004) by involving an extra parameter into a membership certificate during the join phase. The scheme consists of these phases: key generation, join, sign, verify, open and judge.
The sign phase generates a signature $\sigma$ on a message $M \in \{0,1\}^*$ by using a member secret key $gsk = (A, x, y)$ and a group public key $gpk = (G_1, G_2, G_T, e, q, g, k, h = k^{q+1}, g = k^q, g_2 = w^q)$. A signer randomly selects values $\alpha, \beta, r_\alpha, r_\beta, r_x, r_y \in Z_p^*$ and computes values $T_1, T_2, T_3, T_4, R_1, R_2, R_3, R_4$:

$$T_1 = k^\alpha, T_2 = A h_\alpha, T_3 = k^\beta, T_4 = A g^\beta, z = \alpha x + y, R_1 = k^\alpha, R_2 = e(T_2, g_2)^e h(w)^{-e} e(h, g_2)^{-r_\beta}, R_3 = k^\beta, R_4 = h^\alpha g^{-z}.$$ 

The signer computes a challenge $c \in Z_p^*$ using the hash function $H$ and values $\alpha, \beta, s_\alpha, s_\beta, s_x, s_y$ to seal the proof of knowledge of $(\alpha, \beta, x, z)$:

$$c = H(M, T_1, T_2, T_3, T_4, R_1, R_2, R_3, R_4).$$

The signer outputs the signature $\sigma = (T_1, T_2, T_3, T_4, c, s_\alpha, s_\beta, s_x, s_y)$. All pairings can be precomputed and cached like in the BBS scheme. If the signer uses the full precomputation then all values are generated and the parameters in equation (1) are computed in advance. The signer computes only the parameters in equation (1).

In the verification phase, a verifier checks the validity of the signature $\sigma$ generated on the message $M$ by the following:

$$R'_1 = e(T_2, h_1)^{e c} e(w, h_0)^{-s_\alpha} e(w, h_1)^{-s_\beta}, R'_2 = e(g_2, h_1)^{e c} e(T_2, h_0)^{e (g_1, g_2)} R_1 = u^s T_1^{-c}, R_3 = g^d w^s T_3^{-c}, c = H(M, T_1, T_2, T_3, R'_1, R_2, R'_3).$$

If $c$ is equal to restored $c'$, then the verifier accepts the signature and rejects otherwise. All pairings with static values $e(w, h_0), e(w, h_1), e(g_1, h_1)$ can be precomputed in advance. The pairings $e(T_2, h_1)$ and $e(T_2, h_0)$ can be collapsed into one pairing operation.

### 3.2 Non-pairing based group signature schemes

In this subsection, we briefly describe non-pairing anonymous signature schemes such as the ACJT scheme (Ateniese et al., 2000), the CG scheme (Cam Jens and Groth, 2005), the IMSTY scheme (Ishiki et al., 2006) and the HM scheme (Hajny and Malina, 2013).

#### 3.2.1 Ateniese et al. (ACJT) scheme

This group signature scheme (Ateniese et al., 2000) is secure under the strong RSA problem and the decisional Diffie-Hellman problem. The scheme relies on the Fiat-Shamir heuristic (the random oracle model). The scheme consists of five phases: setup, join, sign, verify and open.

The sign phase creates a signature $\sigma$ on a message $M \in \{0,1\}^*$ by using a member secret key $gsk = (x)$, a membership certificate $[A, c]$ and public parameters. A signer randomly selects $w, r_1, r_2, r_3, r_4$ and computes:

$$T_1 = A g^w \mod n, T_2 = g^r \mod n, T_3 = g^r h^w \mod n.$$

The signer computes a challenge $c$ using the hash function $H$ and seals the proof of knowledge:

$$c = H(g, h, y, a_0, a_1, T_1, T_2, T_3, d_1, d_2, d_3, M).$$

The signer outputs the signature $\sigma = (c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$. If the full precomputation is used then all values are generated and the parameters in equation (1) are computed in advance. The signer computes only the parameters in equation (1).

In the verification phase, a verifier checks the validity of the signature $\sigma$ generated on the message $M$ by the following:

$$d'_1 = a_0^T_1^{a_1 - x^2} \mod n, d'_2 = T_2^{a_1 - x^2} \mod n, d_3 = T_3 g^{a_1 - x^2} h^{s_4} \mod n, c' = H(g, h, y, a_0, a_1, T_1, T_2, T_3, d_1, d_2, d_3, d_4, M).$$
If $c$ is equal to restored $c'$, then the verifier accepts the signature and rejects otherwise.

### 3.2.2 Camenisch and Groth (CG) scheme

This group signature scheme (Camenisch and Groth, 2005) is related to the ACJT scheme (Ateniese et al., 2000). The security of the scheme holds under the strong RSA assumption and the decisional Diffie-Hellman assumption. The scheme consists of six phases: key generation, join, sign, verify, open, and user revocation.

#### 3.2.2.1 Key Generation

In the verification phase, a verifier checks the validity of the signature $c$ generated on the message $M$ by the following:

$$v = (aw)^e \cdot r_s \cdot r_e \mod n,$$

where $r_s$ and $r_e$ are random numbers generated and stored as the secret keys of the scheme’s parties. A signer randomly selects $\rho_E, \rho_R, \rho_{r_E}, \rho_{r_R}$, and $\mu_E \in \mathbb{Z}_q$ and computes values $E_0, E_1, E_2, \text{ACOM}, \text{B.COM}, \text{V.COMCipher}, \text{V.COMMPK}, \text{V.COMRev}$:

$$E_0 = [\rho_E]G, E_1 = h_i + [\rho_E]H_2, E_2 = h_i + [\rho_E]H_2.$$

The signer computes a challenge $c$ using the hash function $H$ and values $r_s, r_e, c, \tau, \tau_i, \tau_e, \tau_E$ to seal the proof of knowledge of $(x, s, l, e, E)$:

$$c = H(vk, w, v, U, U_2, U_3, z_x, z_y, z_t, z_r, Z_R).$$

If $c$ is equal to restored $c'$, then the verifier accepts the signature and rejects otherwise.

### 3.2.3 Isshiki et al. (IMSTY) scheme

This group signature scheme (Isshiki et al., 2006) with membership revocation is included in the ISO20008-2 standard (ISO/IEC 20008-2, 2013). The security of this scheme holds under the strong RSA assumption and the decisional Diffie-Hellman assumption. The scheme consists of nine phases: Setup for issuing manager, Setup for setup manager, Setup for the opening manager, join, sign, verify, open, user revocation, and update.

#### 3.2.3.1 Key Generation

In the verification phase, a verifier checks the validity of the signature $c$ generated on the message $M$ by the following:

$$v = (aw)^e \cdot r_s \cdot r_e \mod n,$$

where $r_s$ and $r_e$ are random numbers generated and stored as the secret keys of the scheme’s parties. A signer randomly selects $\rho_E, \rho_R, \rho_{r_E}, \rho_{r_R}$, and $\mu_E \in \mathbb{Z}_q$ and computes values $E_0, E_1, E_2, \text{ACOM}, \text{B.COM}, \text{V.COMCipher}, \text{V.COMMPK}, \text{V.COMRev}$:

$$E_0 = [\rho_E]G, E_1 = h_i + [\rho_E]H_2, E_2 = h_i + [\rho_E]H_2.$$
The signer computes a challenge \( c \) using the hash function \( H \) and seals the proof of knowledge:

\[
c = H(\text{params}, M, A, \bar{A}, G_{PK}, C_1, C_2, C_1, C_2),
\]
\[
C_1, C_2 \subseteq G, z_1 = r_1 - cK_S w_1,
\]
\[
z_2 = r_2 - cK_S w_2, z_3 = r_3 - cK_S w_M, z_S = r_S - cK_S.
\]

The signer outputs the signature \( \sigma = (A, C_1, C_2, c, z_1, z_2, z_1, z_3) \). If the full precomputation is used then all values are generated and the parameters in equation (1) are computed in advance. The signer computes only the parameters in equation (1).

In the verification phase, a verifier checks the validity of the signature \( \sigma \) generated on the message \( M \) by the following:

\[
G_{PK} = A' g_1^{z_1} g_2^{z_2} g_3^{z_3} \mod n,
\]
\[
A = A' g_{PK}^{cK} \mod n, C_1 = C_1' g_1^{z_1} \mod n,
\]
\[
C_2 = C_2' g_2^{z_2} \mod n, c' = H(\text{params}, M, A, \bar{A},
\]
\[
G_{PK}, C_1, C_2, C_1, C_2).
\]

If \( c = c' \), then the verifier accepts the signature and rejects otherwise.

### 4 Theoretical evaluation of anonymous digital signature schemes

In this section, we present the theoretical evaluation of seven ADS schemes (the BBS scheme (Boneh et al., 2004), the ACJT scheme (Ateniese et al., 2000), the DP scheme (Delerablée and Pointcheval, 2006), the HLCCN scheme (Hwang et al., 2011), the CG scheme (Camenisch and Groth, 2005), the IMSTY scheme (Isshiki et al., 2006), the HM GS scheme (Hajny et al., 2013) and two recent group signature schemes with linkability (the HCCN scheme (Hwang et al., 2011), the EH scheme (Emura and Hayashi, 2015)). Table 1 shows the number of operations in sign and verification phases, the length of signatures, revocation mechanisms and complexity that is presented as the number of algorithms/phases used in each scheme. On one hand, we consider pre-caching operations such as bilinear pairing operations of two constant values. These operations can be precomputed once and their results are not variable if different messages are processed. On the other hand, we do not consider the full pre-computation of dynamic parameters (random values) and precomputed coupon techniques. Further, the sign and verification phases are evaluated without the influence of a number of revoked users in the systems (i.e., revocation list is empty) and some private key/credential revocation approaches do not influence the verification. Nevertheless, the impact of the revocation in privacy-preserving services is discussed in Section 6.

In Table 1, we denote a pairing operation as \( P \), exponentiation as \( E \), multiplication as \( M \), addition (subtraction) as \( A \) and a hash function as \( H \). In the pairing-based schemes, the timely execution of exponentiation and multiplication operations can depend on the lengths of elements that are in different groups and fields \( (p, G_1, G_2, G_T) \). The length \( l_{G_2} \) describes the length of a group element \( \in G_1 \) (e.g., 175 bits). The length \( l_{G_2} \) describes the length of a group element \( \in G_2 \) (e.g., 175 bits). The length of a group element \( \in G_T \) is computed as \( k \cdot l_{G_2} \), e.g., 1050 bits, where \( k \) is embedded degree (e.g., \( k=6 \)). \( l_p \) denotes the length of an element in modulo \( p \) (e.g., 170 bits). In the non-pairing based schemes, \( l_n \) denotes the length of the RSA modulo \( n \) (e.g., 1024 bits). \( l_e \) denotes the length of the scalars of various lengths less than \( n \) (e.g., \(< 1024 \) bits). \( l_c \) denotes the length of the hash modulus (e.g., 160 bits). \( l_e \) denotes the length of an elliptic curve element (e.g., 163 bits). The total lengths of signatures depend on the security level chosen.

The most efficient sign phase is provided by the HM scheme (Hajny et al., 2013) which needs to perform only nine exponentiations, 10 modular multiplication, four addition and 1 hash function. The most efficient verification phase is provided by the HM scheme (Hajny et al., 2013) that takes 10 exponentiations, six modular multiplication and 1 hash function. The shortest signature is offered by the HLCCN scheme (Hwang et al., 2011). The HCCN scheme (Hwang et al., 2015) extends the HLCCN scheme (Hwang et al., 2011) and proves its security features (anonymity, traceability, non-frameability, linkability) under a random oracle model. However, the HCCN scheme (Hwang et al., 2015) provides the same number of operations and the length of the signature as the HLCCN scheme (Hwang et al., 2011). Recently, Emura and Hayashi (2015) have proposed group signatures with time-token dependent linking scheme that supports verifier-local revocation and backward unlinkability. The group signature consists of only 6 elements but it is combined with a token that is signed by using a public cryptography scheme, e.g., 3072-bit RSA. Thus, a signer and a verifier have to firstly verify the public cryptography signature of the token during the group signing and verification phases. Further, the group signatures become publicly linkable if signers sign more than once per time period. The EH scheme (Emura and Hayashi, 2015) can be useful for scenarios that require the linkability of signatures in periods, e.g., VANET applications supporting the short-term linkability.

The most complex schemes are the EH scheme (Emura and Hayashi, 2015) and the IMSTY scheme (Isshiki et al., 2006) that consist of nine algorithms. In contrast, the BBS scheme (Boneh et al., 2004) has only 4 basic phases. More complex schemes usually provide more security capabilities, e.g., signatures linking, open signer identity.

In general, pairing-based ADS schemes have shorter signatures and requires less communication/memory cost than non-pairing-based ADS schemes.

### 5 Practical evaluation of anonymous digital signature schemes

This section presents the performance evaluation of ADS schemes on smartphones. We implement and test all seven schemes described in Section 3. Firstly, we introduce our implementation and measurement setup. Then, we present our experimental results on selected devices.
In our comparison, the basic parameters of the implemented ADS schemes are set to 80-bit security level, i.e., ≥ 1024-bit modulus or ≥ 160-bit curves:

- **BBS (Boneh et al., 2004)**, **DP (Delerablé and Pointcheval, 2006)**, **HLCN (Hwang et al., 2011)** and **HCCN (Hwang et al., 2015)**: These schemes use MNT curve parameters (type D) with an embedded degree $k=6$ and the 175-bit order of curves in our implementation. The lengths of the basic parameters are: $l_{G_1}$ is 175 b, $l_p$ is 170 b, $l_{G_2}$ is 175 b and $l_{G_T}$ is 1050 b.

- **ACJT (Ateniese et al., 2000)**: This scheme uses big integers (the strong RSA problem, the decisional DH problem). The lengths of the basic parameters are: $l_p$ is 512 b, $l_n$ is 1024 b and $l_e$ is 160 b.

- **CG (Camenisch and Groth, 2005)**: The scheme uses big integers (the strong RSA problem, the decisional DH problem). The lengths of the basic parameters are: $l_Q$ is 382 b, $l_p$ is 1024 b, $l_n$ is 2048 b and $l_e$ is 160 b.

- **IMSTY (Isshiki et al., 2006)**: The scheme uses big integers and elliptic curves (the strong RSA problem, the decisional DH problem on elliptic curves).
The lengths of the basic parameters are: \( l_n \) is 1024 b, \( l_l \) is 1024 b, \( l_{ecc} \) is 169 b and \( l_c \) is 160 b.

- **HM GS** (Hajny et al., 2013): The scheme uses big integers (the DL problem). The lengths of the basic parameters are: \( l_n \) is 1024 b, \( l_l \) is 160 b, \( l_r \) is 360 b and \( l_c \) is 160 b.

The lengths of other parameters are set in accordance with the recommended lengths that are described in the original papers of the schemes. Devices used in our experimental measurement are specified in Table 2. Mobile I. represents less powerful smartphones (Nexus i9250). Mobile II. represents more powerful smartphones (Nexus 5). PC represents service providers’ servers. We assume that these servers can be more powerful in practice.

All implemented schemes work with a short 80-bit message. The lengths of produced signatures depend on the schemes. We measure the execution times of sign and verification phases that are computed on mobiles and PC (specifications are in Table 2). The execution times are computed as mean values from 10 or 100 measurements (10 – mobiles, 100 – PC).

### Table 2  Technical specifications of devices used

<table>
<thead>
<tr>
<th>Device</th>
<th>Processor</th>
<th>RAM size</th>
<th>Storage size</th>
<th>Operation system</th>
</tr>
</thead>
<tbody>
<tr>
<td>mobile I.</td>
<td>2x 1200 MHz</td>
<td>1024 MB</td>
<td>16 GB</td>
<td>Android 4.2</td>
</tr>
<tr>
<td>(Nexus i9250)</td>
<td>32-bit ARM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mobile II.</td>
<td>4x 2260 MHz</td>
<td>2 GB</td>
<td>16 GB</td>
<td>Android 5.1</td>
</tr>
<tr>
<td>(Nexus 5 LG)</td>
<td>32-bit ARM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>8x 2.53 GHz</td>
<td>8 GB</td>
<td>500 GB</td>
<td>Windows 7</td>
</tr>
<tr>
<td></td>
<td>64-bit Intel</td>
<td></td>
<td></td>
<td>Xeon</td>
</tr>
</tbody>
</table>

#### 5.2 Experimental results

In this subsection, the experimental results of the implemented schemes are presented. Figure 4 depicts the performance of the pairing-based ADS schemes on mobile I. The performance of these schemes on mobile II is shown in Figure 5. The most efficient pairing-based scheme is the DP scheme. Its signing with the pairing precomputation takes around 8 s on mobile II. Its verification of the one signature takes about 24 s on mobile II and the verification with the pairing precomputation takes...
Evaluation of anonymous digital signatures for privacy-enhancing mobile applications

Figure 6 Performance evaluation of non-pairing-based ADS schemes on mobile I (see online version for colours)

Figure 7 Performance evaluation of non-pairing-based ADS schemes on mobile II (see online version for colours)

about 9.8 s. The HLCCN scheme that provides the shortest signature needs more time for both phases than the schemes BBS and DP, i.e., signing with the pairing precomputation takes 11.748 s and verification takes 29.395 s (11.7 s with the pairing precomputation) on mobile II. The signing with the pairing precomputation improves the efficiency of signing and the runtime is reduced by ca 50%. The full precomputation in signing reduces the runtimes for all schemes. The signer needs only several ms (BBS: 3 ms, HLCCN: 3 ms, DP: 4 ms) because he/she computes one hash function and few modular multiplications and additions. The pairing precomputation and pairing collapsing in the verification can improve runtimes by ca 60%.

Figure 6 shows the performance of the non-pairing-based ADS schemes on mobile I. The performance of these schemes on mobile II is depicted in Figure 7. The most efficient non-pairing-based scheme is the HM GS scheme. The signing in this scheme needs only 15 ms on mobile II and 45 ms on mobile I. The verification of one signature in this scheme needs only 12 ms on mobile II and 44 ms on mobile I. Nevertheless, the sign (verification) phase of the IMSTY scheme needs 8984 ms (9872 ms) on mobile II due to the inefficiency of point multiplication operations in the Android platform. All schemes can apply signing with full precomputation and the runtimes of that signing mode take from several hundreds microseconds to few ms.

Further, we investigate the influence of schemes when the numbers of messages and signatures increase to get insights about the schemes working in systems with several messages/signatures in real time. We measure these results on PC that can represent a server deployed in a privacy-preserving system. Figure 8 shows the runtime of the sign phase on PC when the number of messages increases. The pairing precomputation is used in pairing-based schemes. The concrete values in ms can be found in Table 3. On the PC device, the schemes BBS, HLCCN, CG, IMSTY and HM GS are able to produce 20 signatures within 2 s. The scheme ACJT needs a 9 s to sign 20 messages.

Figure 9 presents the runtime of the verification phase on PC when the number of signatures increases and Table 4 shows the concrete values in ms. The schemes BBS, HLCCN, DP, IMSTY, CG and HM GS are able to verify 100 signatures within 10 s. Nevertheless, the pairing-based scheme BBS, HLCCN, DP must use the pairing precomputation trick.
Without this optimisation, these schemes need from 50 s to 60 s to verify 100 signatures. The non-pairing-based scheme ACJT needs 38.302 s for the verification of 100 signatures.

Table 3  Time [ms] of signing for various number of messages $M$ on PC

<table>
<thead>
<tr>
<th>$M$</th>
<th>BBS</th>
<th>HLCCN</th>
<th>DP</th>
<th>ACJT</th>
<th>CG</th>
<th>IMSTY</th>
<th>HM</th>
<th>GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>93</td>
<td>64</td>
<td>89</td>
<td>459</td>
<td>62</td>
<td>113</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>187</td>
<td>129</td>
<td>180</td>
<td>957</td>
<td>123</td>
<td>203</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>459</td>
<td>323</td>
<td>455</td>
<td>2274</td>
<td>316</td>
<td>472</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>924</td>
<td>644</td>
<td>903</td>
<td>4485</td>
<td>605</td>
<td>913</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1865</td>
<td>1281</td>
<td>1964</td>
<td>8932</td>
<td>1228</td>
<td>1764</td>
<td>323</td>
<td></td>
</tr>
</tbody>
</table>

The practical implementation shows that the Android platform has difficulties with large structures of the group elements and curves and with their computations. This can be caused by restrictions on the Android platforms (32-bit CPU architectures, less RAM and cache memories, etc.) and by the Android garbage collector. Therefore, the non-pairing-based ADS schemes (tens to hundreds ms) are more efficient than pairing-based schemes (several seconds) on smartphones with the Android platform. The non-pairing-based ADS schemes are also more efficient than pairing-based schemes on the PC platforms but the performance gap is smaller than on the Android platform. On the other hand, the pairing-based ADS schemes enable to precompute expensive pairing operations with static parameters that cause the decrease of the runtimes during the signing and verification phases. All schemes enable signing with the full precomputation of parameters that takes only a few milliseconds per one signature. Finally, we note that the runtimes of signing and verification can be affected by various random values generated during each signature.
6 Discussions on the perspective of anonymous digital signatures

In this section, we discuss the perspective of ADS schemes in the data collection, access control and data notification systems. We highlight the basic security and practical requirements (i.e., performance, storage and communication cost) for every system, and we recommend the suitable schemes for every system. We also take into account the cost of revocation process. Our recommendation is based on our practical and theoretical results and is summarised in Table 5.

Table 4 Time [ms] of verification for various number of signatures σ on PC

<table>
<thead>
<tr>
<th>#</th>
<th>BBS</th>
<th>HLCCN</th>
<th>DP</th>
<th>ACJT</th>
<th>CG</th>
<th>IMSTY</th>
<th>HM GS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>237</td>
<td>293</td>
<td>223</td>
<td>396</td>
<td>73</td>
<td>117</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>330</td>
<td>389</td>
<td>317</td>
<td>795</td>
<td>141</td>
<td>204</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>609</td>
<td>681</td>
<td>599</td>
<td>1974</td>
<td>343</td>
<td>513</td>
<td>82</td>
</tr>
<tr>
<td>10</td>
<td>1074</td>
<td>1157</td>
<td>1060</td>
<td>3914</td>
<td>670</td>
<td>944</td>
<td>167</td>
</tr>
<tr>
<td>15</td>
<td>1539</td>
<td>1637</td>
<td>1525</td>
<td>5867</td>
<td>1021</td>
<td>1363</td>
<td>255</td>
</tr>
<tr>
<td>20</td>
<td>2004</td>
<td>2098</td>
<td>1990</td>
<td>7882</td>
<td>1350</td>
<td>1848</td>
<td>344</td>
</tr>
<tr>
<td>50</td>
<td>4794</td>
<td>4948</td>
<td>4731</td>
<td>19440</td>
<td>3379</td>
<td>4373</td>
<td>832</td>
</tr>
<tr>
<td>100</td>
<td>9444</td>
<td>9797</td>
<td>9332</td>
<td>38302</td>
<td>6661</td>
<td>8784</td>
<td>1651</td>
</tr>
</tbody>
</table>

6.1 Perspective of anonymous digital signatures in online and offline data collection systems

Data collection systems represent the many-to-one communication model. These systems do not usually require some advanced revocation mechanisms or revocation lists. In some services, the online systems collect messages in short time periods (e.g., real time monitoring services, healthcare monitoring,...). Therefore, the node has to create the signature in a short time (e.g., ≤150 ms). For this scenario, the suitable schemes are the HM GS and CG schemes that satisfy the online monitoring by fast sign and verification phases on recent smartphones.

Further, there are systems (e.g., smart grid consumption data collection) which collect data in longer periods, i.e., hours, days. In these offline (non-real-time) systems with a huge number of nodes (signers), the signed messages are sent to a central server which has to verify many messages. The servers are usually more powerful than signer devices but the length of the signature should be as short as possible in order to mitigate traffic congestion in communication and save storage space. Therefore, the suitable scheme should produce short signatures and be efficient. The pairing based schemes, which provide shorter signatures (∼1300 – 2000 bits) than non-pairing-based schemes (∼6000 – 8000 bits), can be more suitable in this scenario. Nevertheless, the optimisation techniques and tricks such as pairing collapsing and precomputation must be employed to make signing and verification efficient. Moreover, some pairing-based schemes usually enable to use the batch verification that reduces the number of pairing operations to a constant number. Existing advanced software and hardware optimisations, which significantly reduce the time of pairing and point multiplications, make pairing based schemes (e.g., the HLCCN scheme (Hwang et al., 2011) and the EH scheme (Emura and Hayashi, 2015)) useful for offline data collection systems that require short signatures to reduce communication and storage cost.

6.2 Perspective of anonymous digital signatures in access control systems

In access control systems, a client usually sends the response message that contains also the signature of a challenge message from a server (a verifier). On one hand, access control systems do not have high communication and storage cost. On the other hand, the clients usually require a fast authentication process. The signature creation on the client side with a smartphone and the verification on a server together with message communication must take a practical time (e.g., up to 3 s). Moreover, ADS must offer advanced security properties such as an immediate user revocation (i.e., the revocation list), the non-collusions of malicious users and so on. Hence, suitable schemes from our evaluation are the ACJT scheme, the CG scheme and the HM GS scheme that also allow checking revoked users on the verifier side. Nevertheless, the process of checking the revoked users requires some additional operations on the verifier side. For example, the HM GS scheme requires one exponentiation operation in modulo n per one revoked user in the revocation list. If the revocation list increases by new revoked users then the time of authentication process increases too. In order to prevent the slow authentication process, the credentials/private keys should be updated which enable to erase the revocation list.

The pairing-based schemes and the IMSTY scheme can be suitable only if the full precomputation signing mode is applied and the verifier uses a strong device such as PC. Moreover, pairing-based schemes must be enhanced by a revocation phase that also requires some additional operations on the verifier side.

6.3 Perspective of anonymous digital signatures in data notification systems

The data notification systems represent the many-to-many communication model, e.g., VANETs. In these systems, users send messages to other users. Some services require a very short computation overhead (e.g., ≤300 ms) for fast reactions in some safety applications, e.g., break alerts in VANETs. The verification and signature creation must be as fast as possible but the revocation property is not so important in these services. In this scenario, the suitable schemes are the HM GS scheme and the CG scheme. These schemes are very efficient in signing and verification. For example, the verifier (PC) with the HM GS scheme can verify about 20 messages in real time if the runtime of cryptographic overhead must be ≤300 ms. The schemes also provide revocation mechanisms (i.e., revocation lists, credential updates) that are required in some privacy-preserving data notification systems.
Table 5  Suitability of anonymous digital signatures in application scenarios

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Online data collection systems</th>
<th>Offline data collection systems</th>
<th>Access control systems</th>
<th>Data notification systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBS (Boneh et al., 2004)</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>DP (Debrabant and Pointcheval, 2006)</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>HLCCN (Hwang et al., 2011)</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>EH (Emura and Hayashi, 2015)</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>ACJT (Ateiiese et al., 2000)</td>
<td>★</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>CG (Camenisch and Groth, 2005)</td>
<td>★★★</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
<tr>
<td>IMSTY (Isshiki et al., 2006)</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★★★</td>
</tr>
<tr>
<td>HM GS (Hajny et al., 2013)</td>
<td>★★★</td>
<td>★</td>
<td>★★★</td>
<td>★★★</td>
</tr>
</tbody>
</table>

(★ – not suitable; ★★★ – conditionally suitable; ★★★★ – suitable).

7 Conclusions

In this work, we evaluate and compare seven ADS schemes. Our experimental implementation and measurement show that non-pairing-based ADS schemes are more efficient than pairing-based ADS schemes on smartphones and PC in the main important phases such as signing and verification. Nonetheless, the signing phase with the optimisation tricks (full precomputation) takes only a few ms for each scheme on smartphones with the Android platform. The schemes that provide efficient signing and verification are perspective for systems providing access control, online data collection and data notification. The offline data collection systems with many nodes require short lengths of signatures. The pairing-based ADS schemes usually provide short signatures. Nevertheless, these schemes need several seconds in their sign and verification phases if optimisation tricks are not implemented. The revocation of users is also important for some scenarios. For example, the access control systems usually require the immediate rejection of revoked users. This revocation mechanism is offered by schemes that use revocation lists such as the HM GS scheme, the CG scheme and the modified ACJT scheme. These schemes need only several hundred milliseconds on smartphones. Our results also show that the pairing-based ADS schemes using optimisation tricks such as the pairing precomputation can be implemented into some privacy-preserving services using smartphones such as offline data collection and access control.

Acknowledgement

Their research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

References


