Efficient Group Signatures with Verifier-local Revocation Employing a Natural Expiration

Lukas Malina, Jan Hajny and Zdenek Martinasek

Department of Telecommunications, Brno University of Technology, Technicka 12, Brno, Czech Republic

Keywords: Batch Verification, Cryptography, Efficiency, Group Signatures, Privacy, Pseudonymity, Time-bound Secret Key.

Abstract: This paper presents a novel proposal of group signatures with verifier-local revocation employing a natural expiration to ensure an efficient verification of signatures and a revocation check. Current group signatures have an expensive verification phase which takes several pairing operations and checks a long-sized revocation list, especially, if a large number of users are in the group. Generally, the revocation list grows linearly every time when a new revoked user is added into the list unless group parameters and keys are not reinitialized. Nevertheless, the reinitialization is not feasible and burdens the communication overhead in many communication systems. In these schemes, the verification of several signatures with the long-sized revocation list takes too much time. Our proposed group signature scheme offers the more efficient verification phase which employs the revocation list that is reduced in time by a natural expiration of group member secret keys. Due to an optimization in the verification phase, our scheme is more efficient than related solutions.

1 INTRODUCTION

Group signatures can be used in many privacy-preserving services and authentication schemes. A user, who is a member of a group, can sign a message behalf of the group and send the message anonymously to a verifier. Since the first scheme of a group signature is introduced in (Chaum and Van Heyst, 1991), many of group signature schemes have been proposed with various attributes and different ways how to revoke group members. In this paper, we aim to revocation in group signatures which can be divided into three main mechanisms. The first method revokes members by the reinitialization of group public key and sending it to all unrevoked members which must recomputate group member secret keys. This method burdens communication and adds computational operations anytime when a member is added or revoked. The second mechanism is based on sending the single public broadcast message to all members without need to recompute secret keys. In this accumulator-based revocation method, e.g. (Camenisch et al., 2009), users must prove their validity proofs called witnesses and that are included in a white-list accumulator (or not present in a black-list accumulator). Verifiers do not need any revocation list. Nevertheless, signers have to keep track of the changes into accumulator and have to be online. The third option how to check the revoked users is to employ a list with revoked users (keys, credentials etc.) maintained by Group Manager (GM). GM sends it to verifiers who must perform a revocation check. This method is called Verifier-Local Revocation (VLR). Since group members have no work with revocation check, this check must be computed by the verifier, such as in (Boneh and Shacham, 2004) and (Hajny and Malina, 2012). VLR solutions provide less interactivity so the signer can be off-line and has no additional computation in comparing with accumulator-based solutions. The drawback of VLR solutions is usually the growth of revocation lists to enormous sizes in a large group. Our work is aimed to immediate revocation which is also suitable for off-line signers in non-large groups. We focus on the verifier-local revocation approach and propose a group signature scheme with VLR employing a natural expiration to reduce the length of revocation list by time. Our proposal focuses on efficiency in the signing phase and the verification phase including the revocation check.

1.1 Related Work

The verifier-local revocation introduced in (Boneh and Shacham, 2004) can be an efficient revocation so-

Malina L., Hajny J. and Martinasek Z..
Efficient Group Signatures with Verifier-local Revocation Employing a Natural Expiration.
DOI: 10.5220/0004600105550560
In *Proceedings of the 10th International Conference on Security and Cryptography* (SECRYPT-2013), pages 555-560
ISBN: 978-989-8565-73-0

lution for signers. In (Nakanishi and Funabiki, 2007), the authors extend a group signature scheme (Boneh and Shacham, 2004) and add Backward Unlinkability (BU). They employ the revocation tokens of revoked members for certain time intervals to ensure that former signatures cannot be linkable if the member is revoked. Since the proposal (Nakanishi and Funabiki, 2007) is proved in the random oracle model, the work (Libert and Vergnaud, 2009) presents the VLR group signature scheme with BU that is proved in the standard model. Nevertheless, the revocation check also costs 1 pairing operation per one revocation token as in (Nakanishi and Funabiki, 2007). To improve computational overhead, one revocation check is reduced from one pairing to one exponentiation in (Chen and Li, 2012). In (Bringer and Patey, 2011), the scheme proposed in (Chen and Li, 2012) is patched to satisfy backward unlinkability, traceability and exculpability in the random oracle model. The work (Camenisch et al., 2010) presents revocation with efficient updates. The validity time of a credentials is encoded into an attribute. Nevertheless, the solution does not support an immediate revocation. In time-critic services, the solution has to be combined with an accumulator solution. The work (Chu et al., 2012) proposes a pairing-based group signature scheme with VLR employing time-bound secret keys and without BU. Each group secret key has an expiration date so the verifier checks the revocation list that excludes expired members. Only one exponentiation is needed to check whether the key is revoked. Nevertheless, the scheme performs seven pairing operations per one message in the verification phase.

1.2 Our Contribution

Our scheme provides standard group signature properties like authenticity, anonymity, data integrity, non-reputation, correctness and one public key. The scheme does not need the reinitialization of parameters and keys of members when a new user is added, revoked or epoch is ended. In contrary to schemes (Nakanishi and Funabiki, 2007), (Libert and Vergnaud, 2009), (Chen and Li, 2012) and (Bringer and Patey, 2011) where time intervals are employed, in our proposal, a Revocation List (RL) is reduced by the natural expiration of secret keys which is convenient for applications where the individual time of group membership expiration is needed. To our best knowledge, only the scheme proposed by Chu et al. 2012 (Chu et al., 2012) uses time-bound secret keys to the natural expiration of these keys. Nevertheless, we propose a scheme which is more efficient in computational overhead than Chu et al. scheme (Chu et al., 2012) by using a different design and employing optimization techniques such as the batch verification used in (Ferrara et al., 2009) and (Malina et al., 2013). Our scheme needs only 8 elements per a revocation token in contrary to 14 elements needed in (Chu et al., 2012). Moreover, to ensure the shorter revocation tokens, we use time offsets in comparing with using date formats in (Chu et al., 2012). According to the initial results, see section 4.2, our scheme has better performance in the verification phase than the current VLR group signatures.

2 BACKGROUND

In this section, the cryptography background and system model are outlined.

2.1 Cryptography Used

Our scheme is based on a group signature scheme proposed by Boneh and Shacham (the BS04 scheme) (Boneh and Shacham, 2004) with verifier-local revocation that ensures anonymity, authenticity, message integrity, non-repudiation, unlinkability and traceability. The scheme uses bilinear maps and is based on the q-SDH problem and Decision Linear problem, which have been described in (Boneh and Shacham, 2004). We modify this scheme to ensure more efficient verification algorithm by a verifier-local revocation with time-bound group member secret keys and batch verification. To make time-bound group secret member keys, we employ the methods called 0-encoding/1-encoding presented in (Chu et al., 2012). The 0-encoding and 1-encoding reduce the *greater than* predicate to *set intersection* predicate by converting a date format in binary string to a value in Z_p.

2.2 System Model

Our system model consists of three parties:

- Group manager (GM). We assume that GM is a trusted party. GM initializes all group signature parameters, one group public key, one group manager secret key and group member secret keys. GM also manages a revocation list which includes revoked users.
- Verifier (V). V checks only signed messages by a group public key and if user is on the revocation list or not.
- User (U). U, who correctly joins into a group, can sign any message by his/her group member secret key and send it to V.

3 PROPOSED SCHEME

In this section, the proposed scheme is outlined. Our scheme consists of five main phases: setup, join, sign, verify and open. Our scheme is based on BS group signature scheme (Boneh and Shacham, 2004) and it is enhanced on the efficient group signature scheme with time-bound secret keys with batch verification.

3.1 Setup

In the setup phase, GM sets group signature parameters, group public key and group manager secret key. Based on the length of the security parameter λ, the group signature parameters $G_1, G_2, g_1, g_2, \psi, e$ are established since $g_1 = \psi(g_2)$ if $e(\psi(g_2), g_1) \neq 1$. GM generates the group manager secret key $gmsk = (\gamma)$ where $\gamma \overset{R}{\leftarrow} Z_p$. The group public key $gpk = (g_1, g_2, w)$ is published where $w = g_2^{\gamma}$.

3.2 Join

In the join phase, the *i*-th user U$_i$ joins into a group which is managed by a group manager GM as follows:

- Based on the variable values such as the length of revocation list, the reputation of U$_i$ etc., the group manager decides about the duration of expiration date τ_i for the group member secret key gsk_{U_i}. GM encodes the expiration date τ_i by the 1-Encoding: $\{\tau_{ij}\}_{j\in[1,l]} \leftarrow$1-Enc$(\tau_i)$ where l is the length of date format. For $(j = 0; j \leq l; j++)$, GM computes $A_{ij} = g_1^{\frac{1}{\tau_{ij}x_i+\gamma}}$, where $x_{ij} \overset{R}{\leftarrow} Z_p^*$ and $\tau_{ij}x_{ij} + \gamma \neq 0$. GM sends user's group member secret key $\tau_i, \{A_{ij}, x_{ij}\}$, the group public key and public parameters via secured connection to user (e.g. via TLS). The revocation token $\tau_i, \{x_{ij}\}$ is saved.
- U$_i$ encodes the expiration date τ_i by the 1-Encoding: $\{\tau_{ij}\}_{j\in[1,l]} \leftarrow$1-Enc$(\tau_i)$ and checks $e(A_{ij}, w^{\tau_{ij}} g_2^{x_{ij}}) = e(g_1, g_2)$ for each $j \in \{1, 2, ..., l\}$ if gsk_{U_i} is valid.

3.3 Signing

Every user U$_i$ who wants to send a new message to a verifier has to sign the message. Every U$_i$ has a member secret key $gsk_{U_i} = \tau_i, \{A_{ij}, x_{ij}\}$ and a group public key $gpk = (g_1, g_2, w)$. U$_i$ signs a message $M \in (0,1)^*$ and outputs the signature of knowledge $\sigma = (t_{cur}, k, T_1, T_2, c, s_\alpha, s_x, s_\delta, R_2)$ as follows:

1. U$_i$ checks if his/her gsk_{U_i} is not expired by $t_{cur} < \tau_i$, where t_{cur} is a current date (e.g. a current month or a current date in format 'YYMMDD' as in (Chu et al., 2012)) or the date of the signature expiration. If $t_{cur} \geq \tau_i$, the algorithm halts.
2. The dates are converted into *intersection check* by the 0/1-Encoding: $\{\tau_{ij}\}_{j\in[1,l]} \leftarrow$1-Enc$(\tau_i)$ and $\{t_j\}_{j\in[1,l]} \leftarrow$0-Enc$(t_{cur})$ where l is the length of date format used.
3. The index $k \in \{1, 2, ..., l\}$ is found such that $\tau_{ik} = t_k$ and the pair of A_{ik}, x_{ik} from gsk_{U_i} is selected.
4. U$_i$ chooses random elements $\alpha, r_\alpha, r_x, r_\delta \in Z_p^*$.
5. U$_i$ computes the group signature by the following steps:

   Firstly, U$_i$ sets $(\overline{u}, \overline{v}) = H_0(M, gpk, t_{cur})$, where H_0 is two-dimensional hash function, mapping 0,1* to G_2^2. Then, U$_i$ sets their images in G_1 by $(u, v) = \psi(\overline{u}, \overline{v})$ and computes pseudonyms by

   $$T_1 = u^{x_{ik}}, T_2 = A_{ik}v^{\alpha}, \quad (1)$$

   helper values by

   $$\begin{aligned} \delta &= \alpha x_{ik}, R_1 = u^{r_x}, \\ R_2 = e(T_2, g_2)^{-r_x} e(v, g_2)^{r_\delta}\, e(v, w)^{r_\alpha \tau_{ik}} &= \\ e(T_2^{-r_x} v^{r_\delta}, g_2)\, e(v, w)^{r_\alpha \tau_{ik}} & \\ R_3 = T_1^{r_\alpha} u^{-r_\delta}, & \end{aligned} \quad (2)$$

   a challenge value by

   $$c = H(gpk, t_{cur}, M, T_1, T_2, R_1, R_2, R_3), \quad (3)$$

   and response values by

   $$\begin{aligned} s_\alpha &= r_\alpha + c\alpha, \\ s_x &= r_x + cx_{ik}, \\ s_\delta &= r_\delta + c\delta. \end{aligned} \quad (4)$$

6. U$_i$ sends the message M with the signature $\sigma = (t_{cur}, k, T_1, T_2, c, s_\alpha, s_x, s_\delta, R_2)$.

3.4 Verification

The verifier (V) verifies messages received from pseudonymous users. V checks the group signature, the time validity of signature and if a pseudonymous user who signed the received message is not in a revocation list RL.

3.4.1 Individual Verification

Individual verification is performed by V as follows:

1. The time validity of signature is checked by $t_{act} > t_{cur}$, if *yes* then the algorithm halts. To continue the algorithm, the value t_{cur} must be equal or newer than actual date t_{act} measured by verifier.

2. The date t_{cur} is converted into the *intersection check* by the 0-Encoding: $\{t_j\}_{j\in[1,l]} \leftarrow$0-Enc$(t_{cur})$ and by k from signature is found t_k.
3. V restores u,v. . Firstly, V computes $(\overline{u},\overline{v}) = H_0(M,gpk,t_{cur})$, where H_0 is two dimensional hash function, mapping 0,1* to G_2^2. Then, V sets their images in G_1 by $(u,v) = \psi(\overline{u},\overline{v})$.
4. V restores $\overline{R}_1$ and $\overline{R}_3$:
$$\overline{R}_1 = u^{s_x}T_1^{-c}, \overline{R}_3 = u^{-s_\delta}T_1^{s_\alpha}. \tag{5}$$
5. V computes a new control hash c' from the received parameters:
$c' = H(gpk,t_k,M,T_1,T_2,\overline{R}_1,R_2,\overline{R}_3)$.
and checks if $c' = c$. If yes, then V continues with the verification, otherwise the message is inconsistent and is refused.
6. V checks if
$$\begin{aligned} R_2 = e(T_2,g_2)^{-s_x}e(v,w)^{(t_k s_\alpha)} \\ e(v,g_2)^{(s_\delta)}(e(g_1,g_2)e(T_2,w^{t_k})^{-1})^c \\ = e(T_2^{-s_x}v^{s_\delta}g_1^c,g_2)e(v^{s_\alpha}T_2^{-c},w^{t_k}) \end{aligned} \tag{6}$$
7. The signed message is valid if Equations 6 hold.

8. The verification phase continues by a revocation check in the following subsection.

3.4.2 Revocation Check

The verifier opens the actual revocation list RL =$(\tau_i,\{x_{ij}\})$ containing r revoked tokens where $j \in [1,l]$ (l is the length of the date format used) and $i \in [1,r]$ to check if the signed message is received from a revoked or unrevoked user. The revocation check is performed as follows:

- For each i-pair of $\tau_i,\{x_{ij}\}$, V recomputes by the 1-Encoding: $\{\tau_{ij}\} \leftarrow$1-Enc(τ_i) and find index m ($1 \le m \le l$) such that $\tau_{im} = t_k$, selects x_{im} from RL and checks if
$$T_1 = u^{x_{im}}. \tag{7}$$
- If Equation 7 holds then user's signed message will be discarded because the i-th user with x_{im} has been revoked by GM.

If a new user is revoked then GM sends to verifiers the refreshed revocation list. Further, every verifier discards old records with obsolete pairs $\tau_i,\{x_{ij}\}$ to reduce the length of RL.

3.4.3 Batch Verification

If V receives more message in one short period then V verifies the signed messages in one batch. V uses $gpk = (g_1,g_2,w)$ to verify n messages with $\sigma_z = (t_{zcur},k_z,T_{z1},T_{z2},R_{z2},c_z,s_{z\alpha},s_{zx},s_{z\delta})$ for $z = 1,...,n$, and does:

1. V checks the time validity (of signature) by $t_{act} > t_{zcur}$, if *yes* then the algorithm aborts. To continue the algorithm, the value t_{zcur} must be equal or newer than actual date t_{act} measured by verifier.
2. The date t_{zcur} is converted into *intersection check* by the 0/1-Encoding: $\{t_{zj}\}_{j\in[1,l]} \leftarrow$0-Enc$(t_{zcur})$ and by k_z from the signature is found t_{zk}.
3. V restores u_z,v_z. Firstly, V computes $(\overline{u_z},\overline{v_z}) = H_0(M_z,gpk,t_{zcur})$, where H_0 is two-dimensional hash function, mapping 0,1* to G_2^2. Then, V sets their images in G_1 by $(u_z,v_z) = \psi(\overline{u_z},\overline{v_z})$.
4. V restores $\overline{R}_{z1}$ and $\overline{R}_{z3}$:
$$\overline{R}_{z1} = u_z^{s_{zx}}T_{z1}^{-c_z}, \overline{R}_{z3} = u_z^{-s_{z\delta}}T_{z1}^{s_{z\alpha}}, \tag{8}$$
5. V computes a new control hash c'_z from the received parameters:
$c'_z = H(M_z,gpk,t_{zcur},T_{z1},T_{z2},\overline{R}_{z1},R_{z2},\overline{R}_{z3})$.
and checks if $c'_z = c_z$. If yes then V continues with the verification, otherwise the message with the signature is inconsistent and is refused.
6. V randomly selects $\theta_1,\theta_2,...,\theta_n \in Z_p$ with l_b bit, checks the batch if
$$\begin{aligned} \prod_{z=1}^{z=n} R_{z2}^{\theta_z} = e(\prod_{z=1}^{z=n}(T_{z2}^{-s_{zx}}v_z^{s_{z\delta}}g_1^{c_z})^{\theta_z},g_2) \\ e(\prod_{z=1}^{z=n}(T_{z2}^{c_z}v_z^{-s_{j\alpha}})^{\theta_j},\prod_{z=1}^{z=n}(w^{t_{zk}})) \end{aligned} \tag{9}$$
7. The batch with signed messages is valid if Equations 9 hold.
8. V performs the revocation check to ensure that there are no messages from already revoked users.

We can see from Equations 6 and 9 that the individual verification costs 2 pairing operations per one message but the batch verification costs only 2 pairing operations per n messages. In case the batch verification is valid, then all messages from the batch continue are valid. In case the batch verification fails, then the *divide-and-conquer* approach is used to identify the invalid signatures that can be discarded.

3.5 Open

GM stores revocation tokens $\tau_i,\{x_{ij}\}$ of all users. Every correctly signed message M with the group signature σ and group public key can be opened by GM. User index i which is connected with a user ID stored in a database can be revealed by the revocation check.

If the revealed user has still the unexpired group member secret key then GM puts this user onto the revocation list and send refreshed RL to verifiers.

4 SCHEME EVALUATION AND RESULTS

In this section, we evaluate our scheme and compared it with related work. Further, an experimental implementation and initial results of our scheme are outlined. Our scheme is based on the BS04 scheme (Boneh and Shacham, 2004) and inherits all security assumptions of (Boneh and Shacham, 2004). The security proof is not included due to the limitation of the position paper.

4.1 Evaluation and Comparison

We evaluate our solution in the main phases: signing and verification which includes revocation. Table 1 depicts our comparison with related solutions BS04 (Boneh and Shacham, 2004), CLHZ12 (Chu et al., 2012), NF07 (Nakanishi and Funabiki, 2007) and BP11 (Bringer and Patey, 2011). We emphasize that the verification of *n* messages also includes the revocation check of *r* revoked users. Assuming that *p* is a 170-bit prime, the length of elements in G_1 is 171 bits and the length of elements in G_T is 1020 bits. We use the date format for 255 months (21 years) formed in an offset since the setup of system. Then, the date format and index *k* take only 11 bits (8 bits for date, 3 bits for index *k*). Our scheme produces 2059-bit signatures. Comparing with the revocation token used in CLHZ12 (Chu et al., 2012) which has 14 elements, the revocation token has only 8 elements in our scheme. In BP11 scheme (Bringer and Patey, 2011), the size of λ is 80 which afflicts the length of a signature (23301 bits). Due to the batch verification applied in our scheme, the verification takes only 2 pairings per *n* messages.

4.2 Experimental Results

To obtain initial results, we have implemented our proposal as a proof of concept application in JAVA. The main core of our experimental implementation is formed by the group signature scheme that uses the Java Pairing Based Cryptography (jPBC) Library [1]. The implementation employs the MNT curves type D with the embedding degree $k = 6$ and the 171-bit order of curves. Our implementation is tested on a machine with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram. In our scheme, the signing of one message takes approx. 120 ms and one verification with empty RL takes 132 ms. The revocation check with one revocation token in the list takes 5.1 ms. In Figures 1 and 2, the performances of the verification phase of our scheme and related schemes are depicted. The Figure 1 shows the performance of verification of 1 signature with growing the number of revoked users. The Figure 2 depicts the performance of verification with the size of RL $|\text{RL}| = 50$ with growing the number of signatures. Figures 1 and 2 confirms that the verification phase in our scheme is more efficient than the verification phase in the related schemes for a variable number of messages and revoked users. Our scheme is about twice more efficient than the CLHZ12 scheme (Chu et al., 2012).

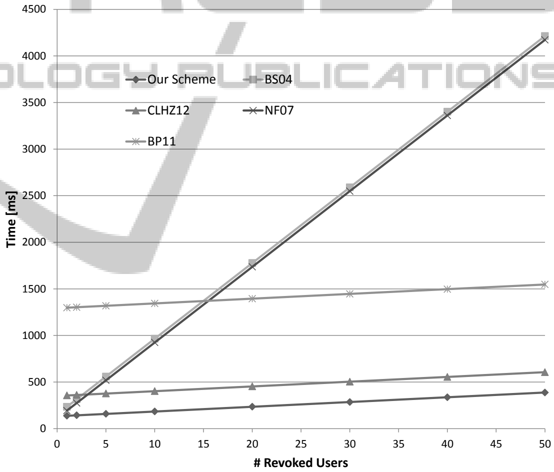

Figure 1: The performance of verification for 1 signature.

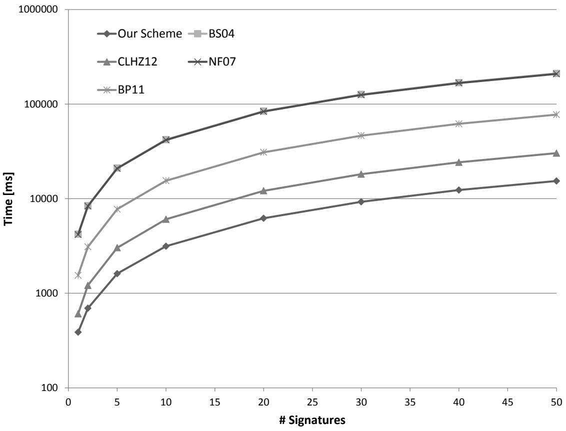

Figure 2: The performance of verification with 50 revoked users.

[1] (available on http://gas.dia.unisa.it/projects/jpbc/index.html)

Table 1: Performance evaluation of VLR group signature schemes - Signing and verification phases.

GS scheme:	Our scheme	BS04 (Boneh and Shacham, 2004)	CLHZ12 (Chu et al., 2012)	NF07 (Nakanishi and Funabiki, 2007)	BP11 (Bringer and Patey, 2011)
Batch:	yes	no	no		
Length of signature:	$2G_1, G_T, 4Z_p$ (2059 bits)	$2G_1, 5Z_p$ (1192 bits)	$4G_1, 5Z_p$ (1549bits)	$3G_1, 6Z_p$ (1533 bits)	$5G_1, \lambda + 6Z_p$ (23301 bits)
Verification of n messages with r revoked users in RL:					
Pairings	**2**	$3n + 2nr$	$7n$	$2n + 2nr$	$1n$
Exponentiation	$10n + 1nr$	$6n$	$13n + 1nr$	$6n$	$3n\lambda+1nr+5n$
Multiplication	$9n+1$	$6n+1nr$	$9n$	$6n+1nr$	$2n\lambda+8n$
Signing:					
Pairings	2	2	5	1	1
Exponentiation	8	8	12	7	16
Multiplication	9	9	10	8	$10+\lambda$

5 CONCLUSIONS

We have presented the group signature scheme with VLR using a natural expiration that can be useful for many applications where back unlinkability is not demanded. Our scheme can be applied in services used by the middle-sized groups of users who are off-line. We have employed batch verification to enhance the performance of our scheme in the verification phase. Hence, verifiers are able to check more signatures at once and save their computational overhead. According to our experimental results, our scheme is more efficient than the related schemes in verification for the various number of signed messages or revocation tokens placed in the revocation list. As future work, we would like to include back unlinkability and investigate the impact of natural expiration on revocation check in large groups.

ACKNOWLEDGEMENTS

This work is funded by the Technology Agency of the Czech Republic projects TA02011260 and TA03010818; the Ministry of Industry and Trade of the Czech Republic projects FR-TI4/647 and FR-TI2/220; and by project SIX CZ.1.05/2.1.00/03.0072.

REFERENCES

Boneh, D. and Shacham, H. (2004). Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 168–177. ACM.

Bringer, J. and Patey, A. (2011). Backward unlinkability for a vlr group signature scheme with efficient revocation check. Technical report, Cryptology ePrint Archive.

Camenisch, J., Kohlweiss, M., and Soriente, C. (2009). An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *Public Key Cryptography–PKC 2009*, pages 481–500. Springer.

Camenisch, J., Kohlweiss, M., and Soriente, C. (2010). Solving revocation with efficient update of anonymous credentials. In *Security and Cryptography for Networks*, pages 454–471. Springer.

Chaum, D. and Van Heyst, E. (1991). Group signatures. In *Advances in CryptologyEUROCRYPT91*, pages 257–265. Springer.

Chen, L. and Li, J. (2012). Vlr group signatures with indisputable exculpability and efficient revocation. *International Journal of Information Privacy, Security and Integrity*, 1(2):129–159.

Chu, C., Liu, J., Huang, X., and Zhou, J. (2012). Verifier-local revocation group signatures with time-bound keys.

Ferrara, A. L., Green, M., Hohenberger, S., and Pedersen, M. Ø. (2009). Practical short signature batch verification. In *Topics in Cryptology–CT-RSA 2009*, pages 309–324. Springer.

Hajny, J. and Malina, L. (2012). Anonymous credentials with practical revocation. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, pages 1–6. IEEE.

Libert, B. and Vergnaud, D. (2009). Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *Cryptology and Network Security*, pages 498–517. Springer.

Malina, L., Castellà-Roca, J., Vives-Guasch, A., and Hajny, J. (2013). Short-term linkable group signatures with categorized batch verification. In *Foundations and Practice of Security*, pages 244–260. Springer.

Nakanishi, T. and Funabiki, N. (2007). A short verifier-local revocation group signature scheme with backward unlinkability. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 90(9):1793–1802.