An ID-Based Signcryption Scheme with Compartmented Secret Sharing for Unsigncryption

Graham Enos[1] and Yuliang Zheng[2]

[1]Dept. of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223, genos@uncc.edu
[2]Dept. of Software and Info. Systems, University of North Carolina at Charlotte, Charlotte, NC 28223, yzheng@uncc.edu

Abstract

In this paper the ID-based signcryption scheme of Li, Xin, and Hu is extended to a compartmented scheme. If an organization is partitioned into different compartments, this scheme allows any member of a specific compartment to participate in the unsigncryption; moreover, each member of a compartment has information unique to that individual. This construction is the first (to the authors' knowledge) to combine identity-based encryption, Shamir's threshold scheme, and signcryption into an implementable compartmented sharing scheme.

1 Introduction

In [10], Li, Xin, and Hu describe an ID-based signcryption scheme that uses a bilinear map to accomplish (t, n) shared unsigncryption with the help of Shamir's secret sharing scheme. Here we describe a way to extend Li et. al.'s construction into a *compartmented scheme*. For our compartmented scheme, suppose the organization $\mathcal{O}$ is split into several compartments $\mathcal{C}_i$, $i \in \{1, \ldots, t\}$. In order to unsigncrypt a message sent to $\mathcal{O}$, at least one member of each of the t compartments must participate; without cooperation from each compartment, the message cannot be unsigncrypted. What's more, each member $\mathcal{M}_{ij} \in \mathcal{C}_i$ gets different information, so although any $\mathcal{M}_{ij}$ can participate equally, we don't have all of $\mathcal{C}_i$ doing the exact same thing.

In what follows, we generally follow the terminology and notation of [10], with a few exceptions. Most notably, uppercase letters usually denote elements in an additive group G_1, lowercase letters denote elements in a multiplicative group G_2, Greek letters are used for elements of $\mathbb{F}_q$, and script letters generally denote compartments or members thereof.

2 Preliminaries

Here we briefly discuss the basic tools needed for our scheme, namely

1. Bilinear maps or pairings
2. Shamir's threshold scheme
3. Signcryption
4. Baek & Zheng's zero knowledge proof for the equality of two discrete logarithms based on a bilinear map

We also cite relevant references for readers wishing more in-depth coverage of these interesting topics.

2.1 Bilinear Maps

Here we will discuss bilinear maps in a somewhat general setting, though most commonly authors focus on those taking as input rational points on an elliptic curve over a finite field. Let G_1 be a cyclic group written additively and G_2 be a cyclic group written multiplicatively (with identity element 1) such that both have the same prime order q. A *bilinear map* or *pairing* is a function $\widehat{e} : G_1 \times G_1 \to G_2$ that satisfies the following properties:

1. *Bilinearity.* For any $P, Q \in G_1$ and $\alpha, \beta \in \mathbb{F}_q^*$, we have $\widehat{e}(\alpha P, \beta Q) = \widehat{e}(P, Q)^{\alpha\beta}$
2. *Non-degeneracy.* There exists $P, Q \in G_1$ such that $\widehat{e}(P, Q) \neq 1$; ergo if $\langle P \rangle = G_1$ then $\langle \widehat{e}(P, P) \rangle = G_2$.
3. *Efficient Computability.* For all $P, Q \in G_1$, the pairing $\widehat{e}(P, Q)$ can be computed efficiently (say, in polynomial time).

In the elliptic curve settings, two popular bilinear maps are the Weil pairing and the Tate pairing; see [5, 13, 15] for details. The Weil pairing was used in Boneh & Franklin's scheme in [2] that gave a solution to the problem originally posed by Shamir in [12].

2.2 Shamir's Threshold Scheme

In [11], Shamir developed a simple and elegant method to share a secret piece of information amongst n people such that no less than some threshold value t of them must cooperate to recover that secret. This scheme uses polynomial interpolation over a finite field; if we suppose that the secret piece of information s is encoded as some element of the field, we then construct a random polynomial f of degree $t - 1$ such that $f(0) = s$ (so s is the constant term). If we give the pair $(i, f(i))$ to the ith person in our scheme, for $1 \leq i \leq n$, then via Lagrange interpolation any group of t people can first reconstruct the polynomial f and

then evaluate $f(0)$ to recover s. Furthermore, because no group of $t-1$ or less people will suffice to recover the polynomial, this scheme is information-theoretically secure. For more, see the original paper [11]; [14] extends the idea of secret sharing to multipartite and compartmented schemes, while [3, 7, 8] and [9] discuss some ways to share secrets in various settings. To our knowledge, however, none of these include identity-based encryption and the next topic: signcryption.

2.3 Signcryption

In [16], the author put forth a new idea that combines the steps of digitally signing and encrypting a message—traditionally two separate procedures—that drastically reduces the computational and communication costs involved. Later work extended this idea to include other cryptographically desirable features, such as non-repudiation, public-verifiability, and forward security (see [10]). Recently made into an international standard, signcryption has gained increasing popularity with researchers and implementers alike. For more information, see the original [16]; [17] demonstrates how to implement signcryption using rational points on elliptic curves over finite fields (important candidates for the additive group G_1 in our scheme). More information, including an extensive bibliography, can be found online at `signcryption.org`.

2.4 Baek & Zheng's zero knowledge proof for the equality of two discrete logarithms based on a bilinear map

Per [1] and [10], the zero knowledge proof of membership for the language

$$L_{\mathtt{EDLog}_{P,\widetilde{P}}^{G_2}} \stackrel{\text{def}}{=} \{(x,\widetilde{x}) \in G_2 \times G_2 \mid \log_g x = \log_{\widetilde{g}} \widetilde{x}\}$$

(where $g = \widehat{e}(P,P)$ and $\widetilde{g} = \widehat{e}\left(\widetilde{P},\widetilde{P}\right)$ for generators P and $\widetilde{P}$ of the additive cyclic group G_1) ensure the robustness of our threshold decryption. Provided that the Decisional Diffie-Hellman problem is hard in G_2 and the Computational and Decisional Bilinear Diffie-Hellman Problems are difficult in $(G_1, G_2, \widehat{e})$, the basic idea is as follows: suppose both the Prover and the Verifier receive the tuple $\left(P,\widetilde{P},g,\widetilde{g}\right)$ and the pair $(k,\widetilde{k}) \in L_{\mathtt{EDLog}_{P,\widetilde{P}}^{G_2}}$. Moreover, suppose the Prover knows a secret $S \in G_1^*$ such that $k = \widehat{e}(S,P)$ and $\widetilde{k} = \widehat{e}(S,\widetilde{P})$; then

1. The Prover chooses at random an element $R \in G_1^*$, computes $a = \widehat{e}(R,P)$ and $\widetilde{a} = \widehat{e}(R,\widetilde{P})$, and sends a and $\widetilde{a}$ to the Verifier.
2. The verifier picks $\gamma \in \mathbb{F}_q^*$ at random and sends it to the Prover.
3. The Prover computes $T = R + \gamma S$ and sends it to the Verifier. If (and only if) the two equalities

$$ak^\gamma = \widehat{e}(T,P) \qquad \widetilde{a}\widetilde{k}^\gamma = \widehat{e}(T,\widetilde{P})$$

hold, the Verifier believes that the Prover knows the secret S since

$$\widehat{e}(T,P) = \widehat{e}(R+\gamma S,P) = \widehat{e}(R,P)\widehat{e}(S,P)^{\gamma} = ak^{\gamma}$$

and

$$\widehat{e}(T,\widetilde{P}) = \widehat{e}(R+\gamma S,\widetilde{P}) = \widehat{e}(R,\widetilde{P})\widehat{e}(S,\widetilde{P})^{\gamma} = \widetilde{a}\widetilde{k}^{\gamma}$$

For more, including how to adapt the above into a non-interactive zero knowledge proof, see [1].

3 The Proposed Compartmented Scheme

Suppose we have an organization $\mathcal{O}$ consisting of n people split into t compartments $\mathcal{C}_i$, each consisting of members $\mathcal{M}_{ij}$. In addition, we have a Private Key Generator ($\mathcal{P}$) who acts as the trusted authority and a sender Alice ($\mathcal{A}$) who wishes to send a message to the compartments $\mathcal{C}_i \subset \mathcal{O}$. There are four stages: **Setup**, **Extraction**, **Signcryption**, and **Unsigncryption**.

3.1 Setup

$\mathcal{P}$ first chooses two groups of large prime order q: an additive group G_1 and a multiplicative group G_2. $\mathcal{P}$ also picks a generator P of G_1, a bilinear map $\widehat{e} : G_1 \times G_1 \to G_2$, and four hash functions

$$\begin{aligned} H_1 &: \{0,1\}^* \to G_1 \\ H_2 &: G_2 \to \{0,1\}^* \\ H_3 &: \{0,1\}^* \times G_2 \to \mathbb{F}_q^* \\ H_4 &: G_2 \times G_2 \times G_2 \to \mathbb{F}_q^* \end{aligned}$$

Finally, $\mathcal{P}$ chooses a secret master key $s \in \mathbb{F}_q^*$, computes $P_{pub} = sP$, and publishes the tuple

$$(G_1, G_2, n, \widehat{e}, P, P_{pub}, H_1, H_2, H_3, H_4, E, D)$$

where E and D are the encryption and decryption steps of some fast symmetric key cipher (like AES; see [6]).

3.2 Extraction

In what follows, given an ID (identifying information considered as a bit string), the public key $\mathcal{P}$ generates for that ID is $Q_{ID} = H_1(ID)$, the private signcryption key is $S_{ID} = s^{-1}Q_{ID}$, and the private decryption key is $D_{ID} = sQ_{ID}$.

Since $\mathcal{P}$ uses $ID_{\mathcal{O}}$ to compute $Q_{\mathcal{O}}$, $S_{\mathcal{O}}$, and $D_{\mathcal{O}}$ and wishes to pass information to each $\mathcal{C}_i$ in such a way that some cooperation is required to put $D_{\mathcal{O}}$ back together, she randomly picks $R_k \in G_1^*$, $k \in \{1, \ldots, t-1\}$, and constructs a function $f : \{0,1\}^* \to G_1$ via $f(u) = D_{\mathcal{O}} + \sum_1^{t-1} u^k R_k$ (treating u as a binary expansion of some positive integer). Then, for each $\mathcal{C}_i \subset \mathcal{O}$, $\mathcal{P}$:

1. Computes $D_i = f(ID_i)$, the private decryption key for $\mathcal{C}_i$
2. Computes $y_i = \widehat{e}(D_i, P)$, the public verification key for $\mathcal{C}_i$
3. For each $\mathcal{M}_{ij} \in \mathcal{C}_i$, $\mathcal{P}$:
   (a) Chooses a random $\mu_{ij} \in \mathbb{F}_q^*$
   (b) Privately sends $\mathcal{M}_{ij}$ the triple $(D_i, P_{ij}, y_{ij}) = \left(D_i, (1+\mu_{ij})D_i, y_i^{\mu_{ij}}\right)$
4. Finally, $\mathcal{P}$ publishes the table

$$\begin{aligned}\{(ID_i, y_i, \{(ID_{ij}, y_{ij})\}\} = (&ID_1, y_1, (ID_{1,1}, y_{1,1}), (ID_{1,2}, y_{1,2}), \dots \\ &(ID_2, y_2, (ID_{2,1}, y_{2,1}), (ID_{2,2}, y_{2,2}), \dots \\ &\vdots\end{aligned}$$

3.3 Signcryption

To send the message m to $\mathcal{O}$, Alice computes the signcrypted text (c, r, S) as follows:

1. She chooses a random $x \in \mathbb{F}_q^*$
2. $k_1 = \widehat{e}(P, Q_{\mathcal{A}})^x$
3. $k_2 = H_2(\widehat{e}(Q_{\mathcal{A}}, Q_{\mathcal{O}})^x)$
4. $c = E_{k_2}(m)$
5. $r = H_3(c, k_1)$
6. $S = (x - r)S_{\mathcal{A}}$

3.4 Unsigncryption

After members $\mathcal{M}_{ij}$ from each of the t compartments $\mathcal{C}_i$ assemble, they first verify Alice's signature; then each $\mathcal{M}_{ij}$ individually:

1. Computes $k_1' = \widehat{e}(S, P_{pub})\widehat{e}(Q_{\mathcal{A}}, P)^r$
2. Accepts Alice's signature if and only if $r = H_3(c, k_1')$

Next, each $\mathcal{M}_{ij}$ picks two random points B_{ij} and $T_{ij} \in G_1$ and uses B_{ij} to certify that they belong to $\mathcal{C}_i$ and T_{ij} to certify their decryption share. While the latter is accomplished in exactly the same manner as in [10], $\mathcal{M}_{ij}$ does the former as follows:

3. Construct credentials κ_{ij} using B_{ij}, where

$$\kappa_{ij} = (\widetilde{P}_{ij}, z_{ij}) = (P_{ij} + B_{ij}, y_{ij}\widehat{e}(B_{ij}, P))$$

4. Send credentials κ_{ij} to each of the other $\mathcal{M}_{k\ell}$

5. Check each of $\mathcal{M}_{k\ell}$'s credentials by testing whether

$$y_k = \frac{\widehat{e}(\widetilde{P}_{k\ell}, P)}{z_{k\ell}}$$

Once everyone's credentials are established, the rest of unsigncryption continues as in [10].

4 Analysis of Scheme

We discuss the effects to correctness, security, and efficiency of the changes we have made to [10]'s original scheme. As such, our analysis is based on that of [10], especially where it makes use of [1]'s zero knowledge proof of membership.

4.1 Correctness

Observe that

$$\begin{aligned}
\frac{\widehat{e}(\widetilde{P}_{k\ell}, P)}{z_{k\ell}} &= \frac{\widehat{e}(P_{k\ell}, P)\widehat{E}(B_{k\ell}, P)}{y_{k\ell}\widehat{e}(B_{k\ell}, P)} \\
&= \frac{\widehat{e}\left((1+\mu_{k\ell})D_k, P\right)}{y_k^{\mu_{k\ell}}} \\
&= \frac{\widehat{e}(D_k, P)\widehat{e}(D_k, P)^{\mu_{k\ell}}}{y_k^{\mu_{k\ell}}} \\
&= \widehat{e}(D_k, P) \\
&= y_k
\end{aligned}$$

So $\kappa_{k\ell}$ does indeed certify that $\mathcal{M}_{k\ell}$ belongs to and can speak for the compartment $\mathcal{C}_k$. The correctness of the rest of our scheme can be proven in exactly the same manner as [10].

4.2 Security

Because the signcryption process in our scheme is the same as in [10] (which in turn is the same as in [4]), our scheme has the same existential unforgeability against chosen plaintext attacks in the random oracle model as those schemes, provided that the Computational Bilinear Diffie-Hellman Problem is difficult in the groups and pairing underlying an implementation of our scheme.

What's more, our scheme doesn't change the level of confidentiality either; assuming the Decisional Bilinear Diffie-Hellman Problem is hard in $(G_1, G_2, \widehat{e})$, our scheme enjoys the same indistinguishability against adaptive chosen ciphertext attacks in the random oracle model. During unsigncryption, no less than t cooperating members of different compartments suffice to recover the key k_2

(and hence the message). Giving different randomly obfuscated versions of the same information to members of the same compartment does nothing to lessen this fact. Recovery of $D_{\mathcal{O}}$ is also computationally infeasible due to the difficulty of inverting the pairing $\widehat{e}$. Finally, the use of Baek and Zheng's zero knowledge proof ensures that each member participating in unsigncryption is protected against the possibility of dishonesty from any of the others.

The public verifiability of our extended scheme remains intact, since any third party can verify the signature via the first two steps of the **Unsigncryption** stage.

We also still keep forward security, since it remains difficult to compute k_2' without $D_{\mathcal{O}}$, even if $S_{\mathcal{A}}$ is leaked.

4.3 Efficiency

With a slight modification to [10]'s notation, let T_p, T_m, and T_e be the computing time required for calculating a pairing, point multiplication, and exponentiation, respectively. Note that our scheme still requires $2T_p + T_m + 2T_e$ for signcryption and $(2t+4)T_p + T_m + (3t-1)T_e$ for $\mathcal{M}_{ij}$, just like the original scheme. The main bottleneck in this scheme is the random point choices performed by $\mathcal{P}$; if we assume that $\mathcal{P}$ has a fast pseudorandom number generator, then the time this takes is essentially $(2n+1)T_m$, just like in [10].

The efficiency picture can be improved, though; instead of having $\mathcal{P}$ choose each $\mathcal{M}_{ij}$'s point, it could instead choose t points and send them to t secondary generators P_i, one for each compartment. These secondary generators can then randomize those points and distribute the relevant information to the members of their respective compartments. Though this doesn't reduce the work involved (and it requires having more trusted authorities, or rather semi-trusted authorities), it does allow our scheme to parallelize one of its major, one-time steps. Hence our scheme lends itself better to implementation using modern computing methods (i.e. parallel computation) than does [10].

5 Conclusion

In this paper we demonstrated how a small modification to Li, Xin, and Hu's scheme ([10]) extends it into a compartmented scheme, allowing a sender to address a message to an organization $\mathcal{O}$ and requiring different compartments $\mathcal{C}_i \subset \mathcal{O}$ to cooperate for the message's recovery. In doing so, we do not lose any of the security or efficiency features of [10]'s scheme—in fact, we can even parallelize one of the main stages. To the authors' knowledge, our scheme is the first that combines identity-based encryption, Shamir's secret sharing, and signcryption into a compartmented sharing scheme that can be implemented with available algorithms and software.

This scheme incorporates a naturally parallelizable step, and is likewise naturally applicable to modern situations. For instance, this scheme could very

easily be used in cloud computing to synchronize information passed to different groups or clusters from a single host. As another example, one could use this scheme for authenticated and signcrypted communication in a business setting; for example, the shared secret could be an expected return message to acknowledge receipt of an important document or the scheduling of an important meeting. In future work, we hope to investigate deeper into questions such as increasing the efficiency of our scheme or reducing the reliance upon the trusted private key generator $\mathcal{P}$.

References

[1] J. Baek and Y. Zheng, *Identity-based threshold decryption*, Public Key Cryptography–PKC 2004 (2004), 262–276.

[2] D. Boneh and M. Franklin, *Identity-based encryption from the weil pairing*, Advances in CryptologyCRYPTO 2001, Springer, 2001, pp. 213–229.

[3] E. Brickell, *Some ideal secret sharing schemes*, Advances in CryptologyEUROCRYPT89, Springer, 1990, pp. 468–475.

[4] S. Chow, S. Yiu, L. Hui, and K. Chow, *Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity*, Information Security and Cryptology-ICISC 2003 (2004), 352–369.

[5] H. Cohen, G. Frey, and R. Avanzi, *Handbook of elliptic and hyperelliptic curve cryptography*, CRC press, 2006.

[6] J. Daemen and V. Rijmen, *The design of Rijndael: AES–the advanced encryption standard*, Springer-Verlag New York Inc, 2002.

[7] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini, *Secret sharing in multilevel and compartmented groups*, Information Security and Privacy, Springer, 1998, pp. 367–378.

[8] S. Iftene, *Compartmented secret sharing based on the chinese remainder theorem*, Tech. report, Cryptology ePrint Archive, 2005.

[9] ———, *General secret sharing based on the chinese remainder theorem with applications in e-voting*, Electronic Notes in Theoretical Computer Science **186** (2007), 67–84.

[10] F. Li, X. Xin, and Y. Hu, *ID-based signcryption scheme with (t, n) shared unsigncryption*, International Journal of Network Security **3** (2006), no. 2, 155–159.

[11] A. Shamir, *How to share a secret*, Communications of the ACM **22** (1979), no. 11, 612–613.

[12] ———, *Identity-based cryptosystems and signature schemes*, Advances in cryptology, Springer, 1985, pp. 47–53.

[13] J.H. Silverman, *The arithmetic of elliptic curves*, vol. 106, Springer Verlag, 2009.

[14] G.J. Simmons, *How to (really) share a secret*, Proceedings on Advances in cryptology, Springer-Verlag New York, Inc., 1990, pp. 390–448.

[15] L.C. Washington, *Elliptic curves: number theory and cryptography*, vol. 50, Chapman & Hall, 2008.

[16] Y. Zheng, *Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption)*, Advances in Cryptology–CRYPTO'97 (1997), 165–179.

[17] Y. Zheng and H. Imai, *How to construct efficient signcryption schemes on elliptic curves*, Information Processing Letters **68** (1998), no. 5, 227–233.