Efficient and Provably Secure Trapdoor-free Group Signature Schemes from Bilinear Pairings

Lan Nguyen and Rei Safavi-Naini

University of Wollongong, Australia
email: [ldn01,rei]@uow.edu.au

Outline

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

Outline

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

Group Signatures

- Participants: Group Manager (GM) and Users.
- GM registers users into the group.
- A group member can anonymously sign on behalf of the group.
- GM can open a signature to find the signer.
- Desirably, GM is split into an issuer and an opener.

Applications

Anonymous Credential systems:

- Organizations and Users.
- An organization gives credentials to users.
- A user can anonymously prove possession of these credentials.
- Each organization is based on a group-signature group.
- Example: idemix (IBM) was developed from ACJT00 group signature scheme (Crypto'00).

Applications

Trust Computing Group: specifications for trusted hardware blocks and software interfaces across platforms.

- Privacy-Preserving Attestation: A device anonymously proves its system components to a remote party.
- Example: Direct Anonymous Attestation (DAA), chips are currently being built.

Others: anonymous authentication, voting and bidding and electronic cash.

Outline

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

Formal Security Models

Bellare-Shi-Zhang (BSZ04) Model

Kiayias-Yung (KY04) Model

Formal Security Model

Bellare-Shi-Zhang (BSZ04) Model

- Participants: an issuer, an opener, users and an adversary.
- Levels of trust for the issuer and the opener: uncorrupt, partially corrupt and fully corrupt.

Formal Security Model

Bellare-Shi-Zhang (BSZ04) Model

- Participants: an issuer, an opener, users and an adversary.
- Levels of trust for the issuer and the opener: uncorrupt, partially corrupt and fully corrupt.
- Procedures: GKg, UKg, Join, Iss, GSig, GVf, Open, Judge

Procedures

- GKg: Set up the group public key gpk, the issuer secret key ik and the opener secret key ok.
- UKg: Set up a personal key pair for a user.
- Join, Iss: The issuer registers a user and the user obtain its membership secret key.
- GSig: Sign a message on behalf of the group.
- GVf: Verify the signature.
- Open: The opener opens the signature to find the signer.
- Judge: Decide if the opener is correct.

Formal Security Model

Bellare-Shi-Zhang (BSZ04) Model

- Participants: an issuer, an opener, users and an adversary.
- Levels of trust for the issuer and the opener: uncorrupt, partially corrupt and fully corrupt.
- Procedures: GKg, UKg, Join, Iss, GSig, GVf, Open, Judge
- Requirements: Correctness, Anonymity, Traceability and Non-frameability

Requirements

- Anonymity: The adversary is given identities of two honest members and a signature generated by one of the members. The probability that the adversary can correctly guess the signer is negligible different from a random guess.
- Traceability: The adversary can not produce a valid signature that the opener can not open to find the correct signer.
- Non-frameability: The adversary can not produce a valid signature that the opener opens and claims an honest member as the signer.

Formal Security Model

Bellare-Shi-Zhang (BSZ04) Model

- Participants: an issuer, an opener, users and an adversary.
- Levels of trust for the issuer and the opener: uncorrupt, partially corrupt and fully corrupt.
- Procedures: GKg, UKg, Join, Iss, GSig, GVf, Open, Judge
- Requirements: Correctness, Anonymity, Traceability and Non-frameability
- Oracles: Open oracle, provided to the adversary in Anonymity requirement definition, takes a signature and output the signer.

Weak Anonymity

- The opener is uncorrupted in Anonymity definition, so it's hard for the adversary to access the Open oracle
- Weak Anonymity = Anonymity - Open oracle
- Group Signatures vs. Public-key Encryption: Anonymity $\sim$ IND-CCA, Weak Anonymity $\sim$ IND-CPA, Open oracle $\sim$ Decryption oracle.
- ACJT00 scheme provides Weak Anonymity, Traceability and Non-frameability, based on proofs by Kiayias and Yung (Cryptology ePrint Archive: Report 2004/076).
- Open question: Does ACJT00 scheme provide Anonymity?

Outline

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

Cryptographic background

Bilinear Pairings: Let $\mathbb{G}_1$ be a cyclic additive group and $\mathbb{G}_M$ be a cyclic multiplicative group, both with order prime p. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_M$ be a bilinear pairing:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_p$
- Non-degeneracy: $e(P_1, P_2) \neq 1$ for some $P_1, P_2 \in \mathbb{G}_1$
- Computability: $e(P, Q)$ computable for all $P, Q \in \mathbb{G}_1$

Assumptions

For a BP tuple $(p, \mathbb{G}_1, \mathbb{G}_M, e)$

- Discrete Log (DL): Given $(Q \in \mathbb{G}_1, xQ)$, compute x $(x \in \mathbb{Z}_p)$.
- Decisional Bilinear Diffie-Hellman (DBDH): Given $P \in \mathbb{G}_1$, distinguish $(aP, bP, cP, e(P,P)^{abc})$ and $(aP, bP, cP, e(P,P)^{r})$ $(a, b, c, r \in \mathbb{Z}_p)$.
- q-Strong Diffie-Hellman (q-SDH): Given $(P \in \mathbb{G}_1, xP, \ldots, x^qP)$, compute a pair $(c \in \mathbb{Z}_p, \frac{1}{x+c}P)$ $(x \in \mathbb{Z}_p)$.

Bilinear Pairing El Gamal Encryption

El Gamal^{BP1}: (K_1, E_1, D_1). Security: providing IND-CPA, assuming DBDH

El Gamal^{BP2}: (K_2, E_2, D_2). Security: providing IND-CCA, assuming DBDH and in the random oracle model. Similar to Pointcheval-Fouque construction, applying Naor-Yung twin paradigm to El Gamal^{BP1} (K_1, E_1, D_1).

- K_2: $(pk_a, sk_a) \leftarrow K_1(l_a)$ and $(pk_b, sk_b) \leftarrow K_1(l_b)$
- E_2: for a message m, the ciphertext is (c_a, c_b, prf), where $c_a = E_1(pk_a, m)$, $c_b = E_1(pk_b, m)$ and a proof prf of $D_1(sk_a, c_a) = D_1(sk_b, c_b)$
- D_2: verifying prf before compute $m = D_1(sk_a, c_a)$

Our two group signature schemes

For a BP tuple $(p, \mathbb{G}_1, \mathbb{G}_M, e)$

- GKg: Group public key $gpk = (P \in \mathbb{G}_1, P_0 \in \mathbb{G}_1, P_{pub} = xP, pk_{enc})$, issuer key $ik = x \in \mathbb{Z}_p$, opener key $ok = sk_{enc}$.

Our two group signature schemes

- GKg: Group public key $gpk = (P, P_0, P_{pub} = xP, pk_{enc})$, issuer key $ik = x$, opener key $ok = sk_{enc}$.
- Join, Iss: User i obtains a random secret x_i, jointly generated by i and the issuer, but known only to i. Both i and the issuer get a_i, S_i, Δ_i, where $e(a_i P + P_{pub}, S_i) = e(P, x_i P + P_0)$ and $\Delta_i = e(P, S_i)$.

Our two group signature schemes

- GKg: Group public key $gpk = (P, P_0, P_{pub} = xP, pk_{enc})$, issuer key $ik = x$, opener key $ok = sk_{enc}$.
- Join, Iss: User i obtains a random secret x_i, jointly generated by i and the issuer, but known only to i. Both i and the issuer get a_i, S_i, Δ_i, where $e(a_iP + P_{pub}, S_i) = e(P, x_iP + P_0)$ and $\Delta_i = e(P, S_i)$.
- GSig: An encryption $c = E(pk_{enc}, \Delta_i)$ and a NZK proof of knowledge of x_i, a_i, S_i so that $c = E(pk_{enc}, e(P, S_i))$ and $e(a_iP + P_{pub}, S_i) = e(P, x_iP + P_0)$ (*).
- Open: Decrypt c to find Δ_i, then i.

Our two group signature schemes

- GKg: Group public key $gpk = (P, P_0, P_{pub} = xP, pk_{enc})$, issuer key $ik = x$, opener key $ok = sk_{enc}$.
- Join, Iss: User i obtains a random secret x_i, jointly generated by i and the issuer, but known only to i. Both i and the issuer get a_i, S_i, Δ_i, where $e(a_i P + P_{pub}, S_i) = e(P, x_i P + P_0)$ and $\Delta_i = e(P, S_i)$.
- GSig: An encryption $c = E(pk_{enc}, \Delta_i)$ and a NZK proof of knowledge of x_i, a_i, S_i so that $c = E(pk_{enc}, e(P, S_i))$ and $e(a_i P + P_{pub}, S_i) = e(P, x_i P + P_0)$ (*).
- Coalition-Resistance: The adversary can adaptively obtain $\{(x_i, a_i, S_i)\}$ satisfying (*), but cannot generate a new (x^*, a^*, S^*) satisfying (*), under the q-SDH assumption.

Our two group signature schemes

- $\mathcal{GS}1$ uses the IND-CCA El Gamal^{BP2} and $\mathcal{GS}2$ uses the IND-CPA El Gamal^{BP1} for the encryption.
- Both provide Traceability assuming q-SDH and Non-frameability assuming DL, in the random oracle model.
- $\mathcal{GS}1$ provides Anonymity, $\mathcal{GS}2$ provides Weak Anonymity, assuming DBDH, in the random oracle model.
- Open Problem: Does $\mathcal{GS}2$ provide Anonymity?

Advantages

- Constant-size signatures and keys.
- Moreover, very short signatures and keys. If ACJT00 scheme uses 1024 bit composite modulus and our schemes uses EC groups of order 170 bit prime, our signature sizes are one third and one half, respectively, of the size in ACJT00 scheme.
- Trapdoor-free. So, different groups can share the same set of parameters.
- Our signatures are based on Perfect ZK proofs without any assumption. ACJT00 signatures are based on Statistical ZK proof under the Strong RSA assumption.

Outline

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

Traceable Signatures

- A traceable signature scheme is a group signature scheme with two added properties: (i) user tracing means given a group member, all his signatures can be traced by a tracer, without using the Open procedure; (ii) signature claiming means a given signature can be provably claimed by its signer.
- Traceable signatures allow more privacy levels for users. For example, tracing all signatures of a misbehaving user can be done without opening signatures and revealing identities of other users.

Traceable Signatures

Kiayias-Yung Model

- Participants: a group manager (GM), users and an adversary

Traceable Signatures

Kiayias-Yung Model

- Participants: a group manager (GM), users and an adversary
- Procedures: Setup, Join, Sign, Verify, Open, Reveal, Trace, Claim, Claim-Verify.
  - Reveal The GM outputs the tracing secret of a user to the tracer.
  - Trace The tracer, with the tracing secret of a user, checks if the signature was signed by the user.
  - Claim A user outputs a proof that he produced a given signature.
  - Claim-Verify A party checks if a claim proof holds.

Traceable Signatures

Kiayias-Yung Model

- Participants: a group manager (GM), users and an adversary
- Procedures: Setup, Join, Sign, Verify, Open, Reveal, Trace, Claim, Claim-Verify.
- Requirements: providing Correctness, and also security against three types of attacks: Anonymity, Misidentification, and Framing (corresponding to Anonymity, Traceability and Non-frameability in the BSZ04 model).

Our Traceable Signature Scheme

- Extended from the group signature $\mathcal{GS}2$
- Similar advantages over the previous scheme: Very short signatures and keys; Trapdoor-free; Based on Perfect ZK proofs.

Summary

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

Summary

- Overview of Group Signatures
  - Descriptions and Applications
- Security Models
  - Procedures, Requirements and Our Reduced Version
- Our Two Group Signature Constructions
  - Cryptographic background, Descriptions, Security and Efficiency analysis
- Our Traceable Signature Scheme

QUESTION, PLEASE?