Group Signatures
Concepts, Applications*, and new Advances**

Anja Lehmann
IBM Research – Zurich

*Zone Encryption with Anonymous Authentication for V2V Communication. J Camenisch, M Drijver, A Lehmann, G Neven, P Towa

**Group Signatures with Selective Linkability. PKC 2019 L Garms, A Lehmann
Roadmap

- Introduction to Group Signatures
  - Setting & Security Properties
  - Schemes
  - Similar Concepts
    - Anonymous Credentials
    - Direct Anonymous Attestation (DAA)
    - Enhanced Privacy ID (EPID)

- Group Signatures & V2X Communication

- Group Signatures with Selected Linkability for V2Cloud
Security property: unforgeability

Important primitive for strong authentication:

- Server-side authentication, certified updates, eID cards, ....

Bad for privacy – “leaks” the identity of the signer

- Membership based online newsportal, vehicle-to-vehicle (V2V) communication, IoT,...
Group Signatures | Naive Approach

- **Privacy**: Doesn’t leak any information about signer
- **Security**: Access to “group” not controlled
  
  No way to reveal signer in case of abuse (bug or feature?)

\[\text{Sign}(sk, m) \rightarrow \sigma\]

\[V_f(pk, m, \sigma) \rightarrow 0/1\]
Group Signatures | High-Level Idea
Chaum & van Heyst’91

- Variants:
  - Static vs dynamic groups
  - Issuer = opener vs dedicated opener
  - Verifiable Opening

- Issues membership credential

- Signed by someone in the Issuer’s group!

- Group public key $gpk$

- Group Manager/Issuer $isk$

- Signed by someone in the Issuer’s group!
Group Signatures | Anonymity

- Signatures don’t leak info about signer
  - Unlinkability of signatures
- Full/CCA anonymity: access to Opener

Corruption Setting
- Issuer corrupt* (if dedicated entity)
- Opener honest

Issuer

JOIN

SIGN

OPEN

issues membership credential

Signed by Alice or Bob?

Signed by the same user?
Group Signatures | Unforgeability (Naïve Approach)

- Forgery = signature on fresh message
- Achievable only if all users are honest  → very weak notion
Group Signatures | Unforgeability

- Realistic model with corrupt users

Is the signature coming from or or ???

Issuer
- Forgery = valid signature that:
  - does not open, or
  - opens to a user that has never joined
Group Signatures | Non-Frameability

- Forgery = valid signature on m that:
  - opens to an honest user U
  - but U has never signed m

Corruption Setting
- Issuer corrupt
- Opener (somewhat) corrupt
### Group Signatures | Security Properties

**Bellare, Shi, Zhang, '05**

<table>
<thead>
<tr>
<th></th>
<th>Anonymity</th>
<th>Traceability</th>
<th>Non-Frameability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuer</td>
<td>Corrupt*</td>
<td>Honest</td>
<td>Corrupt**</td>
</tr>
<tr>
<td>Opener</td>
<td>Honest</td>
<td>Corrupt*</td>
<td>Corrupt</td>
</tr>
</tbody>
</table>

- *Only when Issuer ≠ Opener*
- ** Only for *dynamic* group signatures. Issuer honest in static ones.
- Traceability + Non-frameability = unforgeability
Group Signatures | Schemes

issues membership credential on committed user key

JOIN

proof of knowledge of user key & membership credential

SIGN

OPEN
Choose random usk
\( gsk[i] = (usk, cred) \)

\[ \pi = \text{NIZK}\left\{ usk, upk, cred: Vf(spk, cred, upk) = 1 \land \begin{align*} upk &= PKGen(usk) \\ m &= \tau \rightarrow ssk, spk \end{align*} \right\}(m) \]

\[ m, \sigma = \pi \]

\( SIG. KGen(1^t) \rightarrow ssk, spk \)

\( isk = ssk \)

\( JOIN \)

\( gpk = spk \)

\( upk = PKGen(usk) \)

\( cred = Sign(ssk, upk) \)
**Group Signatures | Schemes**

Choose random usk
\[ gsk[i] = (usk, cred) \]

\[ \pi = \text{NIZK}\left\{ usk, upk, cred: Vf(spk, cred, upk) = 1 \land upk = PKGen(usk) \land C = Enc(epk, upk) \right\}(m) \]

\[ C = Enc(epk, upk) \]

\[ m, \sigma = (\pi, C) \]

ENC. KGen\( (1^t) \) $\rightarrow$ esk, epk

SIG. KGen\( (1^t) \) $\rightarrow$ ssk, spk

**JOIN**

**SIGN**

\[ upk = PKGen(usk) \]

\[ cred = Sign(ssk, upk) \]

\[ isk = ssk \]

\[ gsk[i] = (usk, cred) \]

\[ m, \sigma = (\pi, C) \]

OPEN

\[ gpk = spk, epk \]

\[ upk = Dec esk, C \]

\[ osk = esk \]
Group Signatures | Schemes

- Non-Frameability: PKGen hiding

\[ \text{Choose random usk} \]
\[ \text{gsk[i]} = (\text{usk}, \text{cred}) \]

\[ \pi = \text{NIZK} \begin{cases} \text{usk, upk, cred: Vf(spk, cred)} \\ \text{upk} = \text{PKGen(usk)} \land C = \text{Enc(epk, upk)} \end{cases} \]

\[ C = \text{Enc(epk, upk)} \]

\[ m, \sigma = (\pi, C) \]

- Traceability: Unforgeability of SIG & Soundness of NIZK

\[ \text{ENC. KGen}(1^\tau) \rightarrow esk, epk \]
\[ \text{SIG. KGen}(1^\tau) \rightarrow ssk, spk \]

\[ \text{upk} = \text{PKGen(usk)} \]

\[ \text{cred} = \text{Sign(ssk, upk)} \]

\[ \text{isk} = \text{ssk} \]

\[ \text{JOIN} \]

\[ \text{SIGN} \]

- Anonymity: ZK of NIZK & CCA security of ENC

\[ m, \sigma = (\pi, C) \]

\[ \text{upk} = \text{Dec(esk, C)} \]

\[ \text{osk} = \text{esk} \]

\[ \text{OPEN} \]
**Group Signatures | Schemes**

Bellare, Micciancio, Warinschi’03

- **Sign & Encrypt & Prove** most common approach, mainly differ in signature scheme
  - Signatures on committed messages \( \text{cred} = \text{Sign}(\text{isk}, \text{upk}) = "\text{Sign}(\text{isk}, \text{usk})" \)
  - Efficient proofs of knowledge of a signature
  - Instantiations: CL’01 (strong RSA), CL’04 (LRSW), BBS’04 (q-SDH), PS’16 (q-MSDH-1)

- Opening flexible: verifiable decryption, threshold decryption
- Disadvantage: opening increases signature size, yet is hardly needed

- More compact group signatures: **GetShorty** (Bichsel et al, SCN’10)
  - Join creates user-specific opening secret at Issuer/Opener
  - To open, Issuer/Opener iterates through all opening secrets & test against signature
  - Disadvantage:
    - Opening gets very expensive (feature?)
    - Issuer = Opener (inherently weaker security guarantees)
Roadmap

- Introduction to Group Signatures
  - Setting & Security Properties
  - Schemes
    - Similar Concepts
      - Anonymous Credentials
      - Direct Anonymous Attestation (DAA)
      - Enhanced Privacy ID (EPID)

- Group Signatures & V2X Communication

- Group Signatures with Selected Linkability for V2Cloud
Anonymous Credentials

Envisioned by Chaum in 1981, first full scheme by Camenisch & Lysyanskaya in 2001

- Membership credentials contain user attributes

Attribute-based authentication = group signature on nonce & context wrt attributes-based credential
Anonymous Credentials

- Membership credentials contain user attributes
  - User can **selectively disclose** each attribute
  - User can prove **predicates over the attributes**, e.g., “I'm over 18”
  - **Revocation** of credentials (issuer/verifier-driven)
  - User-controlled linkability via **pseudonyms**
    - **Unlinkable** authentication as default, linkability as an option
  - Construction very similar to group signatures (CL/BBS/PS-based)
Direct Anonymous Attestation (DAA)

- Hardware-based attestation using a Trusted Platform Module (TPM)
  - Secure crypto processor creates, stores, uses cryptographic keys
  - Makes anonymous remote attestations of host status
- Split between host & TPM $\rightarrow$ shift heavy computations to host
- Unlinkability steered via “basename” and pseudonyms. No Opener.
Direct Anonymous Attestation (DAA)

  - RSA-based by Brickell, Camenisch, Chen
  - Developed for Trusted Computing Group (TCG) = industry group that standardizes TPM

- Revised TPM2.0 (2014)
  - Elliptic curve & pairing based
  - Flexible API to support different protocols
  - TPM part & protocols ISO standardized

- Over 500 million TPMs sold

- Standardized DAA has a number of security issues
  - All security models & schemes had issues (ISO scheme is trivially forgeable) \[\text{[CDL16a, CDL16b]}\]
  - TPM interfaces had inherent security problems \[\text{[CCD+17]}\]
  - TPM assumed fully trusted. Subversion-resilient DAA \[\text{[CDL17]}\]
**Enhanced Privacy ID (EPID)**

- DAA-variant used for attestation on Intel’s SGX
  - Without host/TPM split
  - Signature-based revocation

- DAA (and credentials) support **key-based revocation**:

- **Signature-based revocation**:
  - “Bad” signatures
  - Different signer
  - Proof scales linearly in #revoked users

- Revoked keys:
  - Signed with revoked keys?
  - Revocation Authority
  - Relies on exposure of corrupted keys
Comparison

<table>
<thead>
<tr>
<th></th>
<th>Group Signature</th>
<th>Credentials</th>
<th>DAA</th>
<th>EPID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opener</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pseudonyms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revocation</td>
<td></td>
<td>Key-based</td>
<td>Signature-based</td>
<td></td>
</tr>
<tr>
<td>TPM Anchor</td>
<td></td>
<td></td>
<td>All on TPM</td>
<td></td>
</tr>
</tbody>
</table>
Comparison

<table>
<thead>
<tr>
<th></th>
<th>Group Signature</th>
<th>Credentials</th>
<th>DAA</th>
<th>EPID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opener</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pseudonyms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revocation</td>
<td></td>
<td></td>
<td>Key-based</td>
<td>Signature-based</td>
</tr>
<tr>
<td>TPM Anchor</td>
<td>All on TPM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Opener vs. pseudonyms has not only impact on privacy but also on unforgeability
- Every new combination of features requires new security model
- Attributes: can encode validity, i.e., make creds short-lived = alternative to revocation
Comparison

<table>
<thead>
<tr>
<th></th>
<th>Group Signature</th>
<th>Credentials</th>
<th>DAA</th>
<th>EPID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opener</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pseudonyms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revocation</td>
<td></td>
<td></td>
<td>Key-based</td>
<td>Signature-based</td>
</tr>
<tr>
<td>TPM Anchor</td>
<td>All on TPM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Same Core Building-Block: CL/BBS/PS-Signature**

- issues membership credential on committed user key
- proof of knowledge of user key & membership credential
Roadmap

- Introduction to Group Signatures
  - Setting & Security Properties
  - Schemes
  - Similar Concepts
    - Anonymous Credentials
    - Direct Anonymous Attestation (DAA)
    - Enhanced Privacy ID (EPID)

- Group Signatures & V2X Communication

- Group Signatures with Selected Linkability for V2Cloud
Vehicle-to-Vehicle (V2V) Authentication

- Short-range radio communication between vehicles (V2V) and infrastructure (V2I)
  - position, speed,... for collision avoidance, road & traffic conditions
  - first roll-out in 2019(?), expected mandatory in new vehicles in near future

- Requirements:
  - **security**: authenticate real vehicles to exclude attacker trying to disrupt traffic
  - **privacy**: cannot track vehicles by unique identifiers in radio messages

- V2V/V2I (=V2X)
  - low communication bandwidth (300 Bytes max)
  - high message frequency (1-10 msg/vehicle/second)
Current C-ITS Security Architecture

- C-ITS: Cooperative Intelligent Transport Systems
  - Standardization in CEN and ETSI

- C-ITS Platform established by European Commission in 2014
  - Cooperative framework incl. national authorities, C-ITS stakeholders and the Commission
  - Develop a shared vision on the interoperable deployment of C-ITS in the EU
Current C-ITS Security Architecture with Pseudonym CA

- Vehicles receive short-term pseudonym certificates (100/week), switch every 5min
- Authenticate messages via pseudonym certificates

Neither optimal for privacy nor security:
- Pseudonym CA is security/privacy bottleneck & expensive to maintain
- High storage costs for vehicles
- Limited pool of pseudonyms
Group Sigs/Credentials: Optimal Privacy and Security

- Different key ("credential") in each vehicle, can be individually revoked
- Offline authority (or multiple) can de-anonymize signatures

- Vehicles can locally self-certify pseudonyms
  - no server interaction needed
  - optionally limit number of pseudonyms per vehicle/day/...
Group Sigs/Credentials: Optimal Privacy and Security

- Different key ("credential") in each vehicle, can be individually revoked
- Offline authority (or multiple) can de-anonymize signatures

Main challenge: efficiency & bandwidth & revocation
(300 Bytes max, 1-10 sig per vehicle/sec)

- Optionally limit number of pseudonyms per vehicle/day/...
V2X Communication via Group Signatures

- Our approach:
  - Long-term conventional certificate (revocation is easy)
  - Short-lived group membership credentials incl attribute = validity epoch, e.g., week
  - Compact sigs: GetShorty + PS group signatures + attribute

<table>
<thead>
<tr>
<th>Sig Size</th>
<th>Signing</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>2G + 3Zp</td>
<td>1G</td>
<td>1G’ + 2P</td>
</tr>
</tbody>
</table>

BLS381: 176 Byte per signature

Wait .... what are we actually signing?

Wait .... what are we actually signing?
V2X Communication via Group Signatures

- Regular position beacon messages, broadcasted 1–10 times per second
  - Cooperative Awareness Messages (CAMs)
  - Dynamic information: position, speed, and heading
  - Static information: length, width, and sensor accuracy
- Signed with privacy-preserving (group/pseudonym) signature but broadcast in plaintext

Group Signature cannot guarantee privacy when messages are already identifying!
V2X Communication via Group Signatures & Encryption

- Privacy-preserving V2X communication needs encryption!
- New Approach: Zone Encryption with Anonymous Authentication [CDLNT19]
  - Vehicles exchange short-lived & geo-local *symmetric* AE keys
  - Use (compact) group signatures for authenticated key-exchange
  - Send CAMs encrypted with AE keys (w/o group signature)
  - Legitimate vehicles can decrypt, but no passive eavesdropping & mass surveillance
Roadmap

▪ Introduction to Group Signatures
  ▪ Setting & Security Properties
  ▪ Schemes
  ▪ Similar Concepts
    ▪ Anonymous Credentials
    ▪ Direct Anonymous Attestation (DAA)
    ▪ Enhanced Privacy ID (EPID)

▪ Group Signatures & V2X Communication
  ▪ Group Signatures with Selected Linkability for V2Cloud
Vehicle-to-Cloud Communication

- V2Cloud communication: updates, diagnostics, services (e.g., insurance)
  - Less resource critical (via 4/5G, Wifi), less frequent
- Collection of sensor, driver data – general statistics, user-specific services
  - Data usage often not clear at time of collection
  - Requirements: authenticity & privacy
Vehicle-to-Cloud Communication with Group Signatures

- Which variant to control privacy vs utility?
  - **Opening** not suitable – too invasive and inefficient. Might have to open all signatures
  - **User-controlled linkability (pseudonym)** too inflexible:
    - Decision about linkability must be done at the moment the data is disclosed
    - No option to selectively correlate data later on \(\rightarrow\) bad tradeoff between privacy and utility
    - Static pseudonyms allow inference attacks

<table>
<thead>
<tr>
<th>NYM</th>
<th>Usage</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Par6q</td>
<td>4.1l</td>
<td>52 km/h</td>
</tr>
<tr>
<td>yK11s</td>
<td>7.8l</td>
<td>64 km/h</td>
</tr>
</tbody>
</table>
Group Signatures with Selective Linkability [GL19]

- Extends group signatures to allow for selective linkability after the data is collected
  - Data is fully unlinkable and anonymous when its collected
  - Selective subsets can be correlated in a consistent manner later on
  - Linkability is created through a dedicated entity → the converter

Optimal privacy when data is collected while preserving the full utility of the data.
Data is collected in unlinkable, authenticated snippets
- Group signatures with fresh pseudonyms for every message
  → Cloud is assured that only legitimate data gets uploaded & full privacy is preserved
Group Signatures with Selective Linkability | Convert

- Only required sub-sets of the data are made linkable w.r.t. to join-specific pseudonym
- Converter transforms pseudonyms into consistent representation
  - **Obliviousness**: converter learns nothing about pseudonyms / messages it transforms
  - **Non-transitivity**: different conversion requests cannot be linked
Summary

- Group signatures: privacy-preserving authentication
- Many variants & extensions exist:
  - Opener, pseudonyms, attributes, hardware-based, revocation, ...
  - Anonymous Credentials, DAA, EPID
- Defining security for group signatures requires a lot of care
- Group signature cannot guarantee privacy when messages are already identifying!

Thanks! Questions?

anj@zurich.ibm.com