A Revocation Scheme Preserving Privacy

Łukasz Krzywiecki, Przemysław Kubiak,
Mirosław Kutyłowski

Institute of Mathematics and Computer Science
Wrocław University of Technology

INSCRYPT, Beijing 2006

Wrocław
University
of Technology

Revocation problem in broadcasting systems

- broadcast of encrypted data,
- access to data only with a decryption key
- the decryption key shown only to the users that pay for transmission.

Revocation problem in broadcasting systems

- broadcast of encrypted data,
- access to data only with a decryption key
- the decryption key shown only to the users that pay for transmission.

Main problem – removing some number of users from the system:

change the key so that the new key can be decoded only by the non-removed users

Wrocław University of Technology

Goals

Goal 1: low communication – communication overhead due to messages encoding the new key should be minimized,

Goal 2: user anonymity – analysis of data sent does not reveal user's behavior,

the second feature has been neglected so far

Wrocław University of Technology

Revocation via Lagrangian Interpolation in the Exponent

Communication Complexity

Let z be a parameter denoting an upper bound for the number of revoked users.
Then message required to change the key has length $O(z)$.

Message length **does not depend** on the number of users that remain.

Wrocław University of Technology

Initialization

Procedure $\mathrm{Init}_{\mathrm{BE}}$

input the maximum number of revoked users z,

output master secret SK_{BE},
which is a random polynomial $L(x)$ of degree z.

Wrocław University of Technology

Registration of a User

Procedure $\mathrm{Reg}_{\mathrm{BE}}$

input master secret SK_{BE} and a new user u,

output user's u secret share $SK_{u,\mathrm{BE}} = (x_u, L(x_u))$.

Wrocław University of Technology

Encoding a New Key

Procedure $\mathrm{Enc}_{\mathrm{BE}}$

input
- the master secret SK_{BE},
- a new session key K,
- a set of users to be revoked, of cardinality $\leq z$

output so called *enabling block* H.

Construction of H will follow.

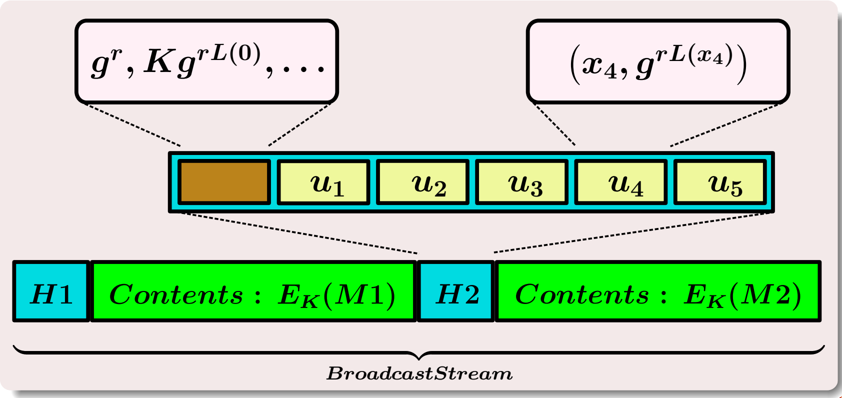
Deriving a new Key

Procedure $\mathrm{Dec}_{\mathrm{BE}}$

input
- the enabling block H,
- user's u secret share $SK_{u,\mathrm{BE}}$,

output session key K, if u is a legitimate user, otherwise *error*.

Enabling block H



Lagrangian Interpolation in the Exponent

Given: $z+1$ pairs $\left(x_u, g^{rL(x_u)}\right)$

then $g^{rL(0)}$ can be reconstructed by Lagrangian Interpolation in the Exponent.

Wroclaw University of Technology

Lagrangian Interpolation in the Exponent

Given: $z+1$ pairs $\left(x_u, g^{rL(x_u)}\right)$

then $g^{rL(0)}$ can be reconstructed by Lagrangian Interpolation in the Exponent.

indeed:

$$g^{rL(0)} = \prod_{0 \le u \le z} (g^{rL(x_u)})^{\lambda_u(0)} \qquad = g^{r \sum_{u=0}^{z} L(x_u)\lambda_u(0)},$$

where $\lambda_u(x) = \prod_{0 \le v \le z, v \ne u} \frac{x - x_v}{x_u - x_v}$,

and g is a generator of a cyclic group G of prime order q.

Wrocław University of Technology

Exclusion Idea

- a key K is encoded as $K \cdot g^{rL(0)}$,

Wrocław University of Technology

Exclusion Idea

- a key K is encoded as $K \cdot g^{rL(0)}$,

- if user u has to be excluded, then the share $\left(x_u, g^{rL(x_u)}\right)$ is in the enabling block,
- exactly z shares are included in the enabling block,

Wroclaw University of Technology

Exclusion Idea

- a key K is encoded as $K \cdot g^{rL(0)}$,

- if user u has to be excluded, then the share $\left(x_u, g^{rL(x_u)}\right)$ is in the enabling block,
- exactly z shares are included in the enabling block,

- a non-excluded user v can construct one more share: $x_v, (g^r)^{L(x_v)}$.
- an excluded user has not enough shares for applying Lagrangian interpolation.

Privacy Threats

Problem

Values x_u are the same in subsequent sessions for user u.

Possible threats from an Adversary

- analyzing activity of the users,
- resolving users' preferences,
- finding behavioral patterns for groups,

Threats for a single user as well as leaking global characteristics of system usage.

Wroclaw University of Technology

Solution Idea - How to Ensure Anonymity

Let users' shares change

according to some random polynomial $x_u(t)$.

- $x_u(t)$ is known only to the broadcaster and user u,

- for each enabling block a random parameter t_ℓ is chosen,
- if u gets excluded, then the enabling block contains value $x_u(t_\ell)$, which does not reveal u.

Wroclaw University of Technology

A Naive Approach – Initialization

Init_{BE}

input the maximum number of revoked users z,

output master secret SK_{BE} which is a polynomial

$$L(t,x) = \sum_{i=0}^{z} \left(a_i(t) \cdot x^i\right) \quad \text{where} \quad a_i(t) = \sum_{j=0}^{\alpha} a_{i,j} t^j$$

Wrocław University of Technology

A Naive Approach – Registration

$\mathrm{Reg}_{\mathrm{BE}}$

input master secret SK_{BE} and a new user index u

output user secret share $SK_u = (x_u(t), L(t, x_u(t)))$.

$x_u(t)$ generated at random,
$L(x_u(t))$ obtained via superposition:

$$L(t, x_u(t)) = \sum_{i=0}^{z} \left(a_i(t) \cdot x_u(t)^i \right) = \sum_{k=0}^{\alpha z} c_k t^k$$

Wrocław University of Technology

An Attack on the Naive Scheme

A malicious user u' takes arbitrary $t_0, t_2, \ldots, t_{\alpha+z\beta}$ and solves linear equation system

$$\begin{cases} L(t_1, x_{u'}(t_0)) & = & \sum_{i=0}^{z}\left(\sum_{j=0}^{\alpha} a_{i,j} t_0^j\right) \cdot (x_{u'}(t_0))^i \\ \vdots & \vdots & \vdots \\ L(t_{\alpha+z\beta}, x_{u'}(t_{\alpha+z\beta})) & = & \sum_{i=0}^{z}\left(\sum_{j=0}^{\alpha} a_{i,j} t_{\alpha+z\beta}^j\right) \cdot (x_{u'}(t_{\alpha+z\beta}))^i \end{cases}$$

Wrocław University of Technology

An Attack on the Naive Scheme

A malicious user u' takes arbitrary $t_0, t_2, \ldots, t_{\alpha+z\beta}$ and solves linear equation system

$$\begin{cases} L(t_1, x_{u'}(t_0)) & = & \sum_{i=0}^{z}\left(\sum_{j=0}^{\alpha} a_{i,j} t_0^j\right)\cdot(x_{u'}(t_0))^i \\ \vdots & \vdots & \vdots \\ L(t_{\alpha+z\beta}, x_{u'}(t_{\alpha+z\beta})) & = & \sum_{i=0}^{z}\left(\sum_{j=0}^{\alpha} a_{i,j} t_{\alpha+z\beta}^j\right)\cdot(x_{u'}(t_{\alpha+z\beta}))^i \end{cases}$$

Adversary breaks the schema

he learns master secret SK_{BE}, i.e. “coefficients” of $L(t, x)$.

Wrocław University of Technology

Our Solution – Initialization

Procedure $\mathrm{Init}_{\mathrm{BE}}$

input the maximum number z of revoked users, and the number z_d of dummy "users",

output master secret SK_{BE}, consisting of polynomials:

$$L(t,x) = \sum_{i=0}^{z+z_d} (a_i(t) \cdot x^i), \quad \text{where} \quad a_i(t) = \sum_{j=0}^{\alpha} a_{i,j} t^j$$

$$S(t) = \sum_{j=0}^{\gamma} s_j \cdot t^j$$

Wrocław University of Technology

Our Solution – Registration

Procedure $\mathrm{Reg}_{\mathrm{BE}}$

input the master secret SK_{BE} and a new user u,

output user's u secret share $SK_u = (x_u(t), P_u(t), g^{Q_u(t)})$,

where

$$P_u(t), Q_u(t)$$

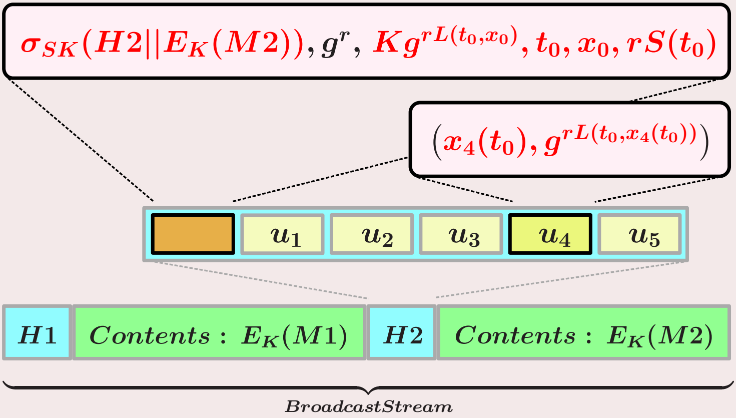
are some polynomials such that

$$L(t, x_u(t)) = \sum_{i=0}^{z+z_d} \left(a_i(t) \cdot x_u(t)^i \right) = P_u(t) + Q_u(t) \cdot S(t).$$

Wroclaw University of Technology

Our Solution – The Enabling Block

Header Construction



A Legitimate User u Computes the Session Key K

First she computes her own share

- $x_u(t_0)$,
- $g^{rL(t_0, x_u(t_0))} = (g^r)^{P_u(t_0)} \cdot \left(g^{Q_u(t_0)}\right)^{rS(t_0)} = g^{rP_u(t_0) + rQ_u(t_0)S(t_0)}$.

Wrocław University of Technology

User u Computes the Session Key K

Given: $z + z_d + 1$ pairs $\left(\psi_u, g^{rL(t_0,\psi_u)}\right)$

Mask $g^{rL(t_0,x_0)}$ can be reconstructed by Lagrangian Interpolation in the exponent,
and K can be derived from $K \cdot g^{rL(t_0,x_0)}$ available in the enabling block.

$$g^{rL(t_0,x_0)} = \prod_{0 \le u \le z+z_d} \left(g^{rL(t_0,\psi_u)}\right)^{\lambda_u(x_0)} = g^{r\sum_{u=0}^{z+z_d} L(t_0,\psi_u)\lambda_u(x_0)},$$

where $\lambda_u(x) = \prod_{0 \le v \le z+z_d, v \ne u} \frac{x-\psi_v}{\psi_u-\psi_v}$ and $\psi_u = x_u(t_0)$ for a real user u, but ψ_u is a random value for a dummy "user".

Wrocław University of Technology

Why the Attack Does Not Work

a malicious user u'

this time has to cope with equation system in the exponent, with unknown $L(t, x)$, $Q_u(t)$, $S(t)$

$$\begin{cases} g^{L(t_1, x_{u'}(t_1))} & = & g^{P_{u'}(t_1)+Q_{u'}(t_1)S(t_1)} & = & ? \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g^{L(t_n, x_{u'}(t_n))} & = & g^{P_{u'}(t_n)+Q_{u'}(t_n)S(t_n)} & = & ? \end{cases}$$

u' does not know the values "?", from headers he knows only $g^{rL(t_i, x_{u'}(t_i))}$, where r is random for each new header.

Wrocław University of Technology

Why the Attack Does Not Work

a malicious user u'

this time has to cope with equation system in the exponent, with unknown $L(t,x)$, $Q_u(t)$, $S(t)$

$$\begin{cases} g^{L(t_1,x_{u'}(t_1))} & = & g^{P_{u'}(t_1)+Q_{u'}(t_1)S(t_1)} & = & ? \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g^{L(t_n,x_{u'}(t_n))} & = & g^{P_{u'}(t_n)+Q_{u'}(t_n)S(t_n)} & = & ? \end{cases}$$

u' does not know the values "?", from headers he knows only $g^{rL(t_i,x_{u'}(t_i))}$, where r is random for each new header.

Getting any of the $L(t,x)$, $Q_u(t)$, $S(t)$ for such a system is a hard problem.

Wrocław University of Technology

Security of the Scheme

- Values $r \cdot S(t_0)$ are present in the header, where r and t_0 are freshly generated for each new header.

Wroclaw University of Technology

Security of the Scheme

- Values $r \cdot S(t_0)$ are present in the header, where r and t_0 are freshly generated for each new header.
- r and $S(t_0)$ mask each other.

Wrocław University of Technology

Security of the Scheme

- Values $r \cdot S(t_0)$ are present in the header, where r and t_0 are freshly generated for each new header.
- r and $S(t_0)$ mask each other.
- If the values could be separated, the system would be broken.
- ...

Further details in the paper.

Wrocław University of Technology

Thank you for your attention!

Wrocław University of Technology

Wrocław University of Technology

... Why the attack does not work

- u' knows $P_{u'}$, hence he might compose a system

$$\begin{cases} L(t_1, x_{u'}(t_1)) - Q_{u'}(t_1)S(t_1) & = & P_{u'}(t_1) \\ \vdots & \vdots & \vdots \\ L(t_n, x_{u'}(t_n)) - Q_{u'}(t_n)S(t_n) & = & P_{u'}(t_n). \end{cases}$$

...Why the attack does not work

- u' knows $P_{u'}$, hence he might compose a system

$$\begin{cases} L(t_1, x_{u'}(t_1)) - Q_{u'}(t_1)S(t_1) & = & P_{u'}(t_1) \\ \vdots & \vdots & \vdots \\ L(t_n, x_{u'}(t_n)) - Q_{u'}(t_n)S(t_n) & = & P_{u'}(t_n). \end{cases}$$

- Denote by $L_u(t)$ the polynomial $L(t, x_u(t)) = \sum_{j=0}^{\alpha+(z+z_d)\beta} c_{u,j} t^j$.

. . . Why the attack does not work

- u' knows $P_{u'}$, hence he might compose a system

$$\begin{cases} L(t_1, x_{u'}(t_1)) - Q_{u'}(t_1)S(t_1) & = & P_{u'}(t_1) \\ \vdots & \vdots & \vdots \\ L(t_n, x_{u'}(t_n)) - Q_{u'}(t_n)S(t_n) & = & P_{u'}(t_n). \end{cases}$$

- Denote by $L_u(t)$ the polynomial $L(t, x_u(t)) = \sum_{j=0}^{\alpha+(z+z_d)\beta} c_{u,j} t^j$.
- Hence u' might "calculate" coefficients of the polynomial $L_{u'}(t) - Q_{u'}(t)S(t)$

...Why the attack does not work

- u' knows $P_{u'}$, hence he might compose a system

$$\begin{cases} L(t_1, x_{u'}(t_1)) - Q_{u'}(t_1)S(t_1) & = & P_{u'}(t_1) \\ \vdots & \vdots & \vdots \\ L(t_n, x_{u'}(t_n)) - Q_{u'}(t_n)S(t_n) & = & P_{u'}(t_n). \end{cases}$$

- Denote by $L_u(t)$ the polynomial $L(t, x_u(t)) = \sum_{j=0}^{\alpha+(z+z_d)\beta} c_{u,j} t^j$.
- Hence u' might “calculate” coefficients of the polynomial $L_{u'}(t) - Q_{u'}(t)S(t)$ $= [L_{u'}(t) + \alpha(t)S(t)] - [Q_{u'}(t) - \alpha(t)]S(t)$

... Why the attack does not work

- u' knows $P_{u'}$, hence he might compose a system

$$\begin{cases} L(t_1, x_{u'}(t_1)) - Q_{u'}(t_1)S(t_1) & = & P_{u'}(t_1) \\ \vdots & \vdots & \vdots \\ L(t_n, x_{u'}(t_n)) - Q_{u'}(t_n)S(t_n) & = & P_{u'}(t_n). \end{cases}$$

- Denote by $L_u(t)$ the polynomial $L(t, x_u(t)) = \sum_{j=0}^{\alpha+(z+z_d)\beta} c_{u,j}t^j$.
- Hence u' might "calculate" coefficients of the polynomial $L_{u'}(t) - Q_{u'}(t)S(t)$ $= [L_{u'}(t) + \alpha(t)S(t)] - [Q_{u'}(t) - \alpha(t)]S(t) = P_{u'}(t)$.

. . . Why the attack does not work

- u' knows $P_{u'}$, hence he might compose a system

$$\begin{cases} L(t_1, x_{u'}(t_1)) - Q_{u'}(t_1)S(t_1) & = & P_{u'}(t_1) \\ \vdots & \vdots & \vdots \\ L(t_n, x_{u'}(t_n)) - Q_{u'}(t_n)S(t_n) & = & P_{u'}(t_n). \end{cases}$$

- Denote by $L_u(t)$ the polynomial $L(t, x_u(t)) = \sum_{j=0}^{\alpha+(z+z_d)\beta} c_{u,j} t^j$.
- Hence u' might "calculate" coefficients of the polynomial $L_{u'}(t) - Q_{u'}(t)S(t)$ $= [L_{u'}(t) + \alpha(t)S(t)] - [Q_{u'}(t) - \alpha(t)]S(t) = P_{u'}(t)$.
- Note that almost any $\alpha(t)$ such that $\deg \alpha \leq \deg Q_{u'}$ does not change the degree of "polynomial" $g^{Q_{u'}}$ known to u'. Hence almost each of the $|p|^{1+\deg Q_{u'}}$ possibilities is a right solution for the above system.