




Establishing Bilateral Anonymous Communication in Open Networks


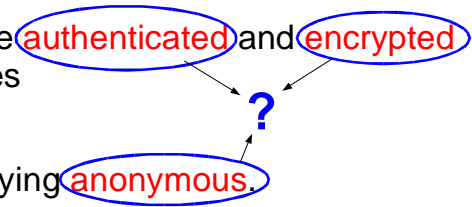
Thomas Demuth
Department of Communication Systems
University of Hagen
Germany



Intention of the Talk

To show, how it is possible, that two partners can

- › initialise a communication and
- › exchange **authenticated** and **encrypted** messages
- › while staying **anonymous**.



Anonymity and Security

Anonymity covers different aspects of security:

- › **Untraceability**
 - › of messages
 - › of communicating partners
- › **Unlinkability**
 - › of messages
- › **Confidentiality**
 - › of messages



Mix Networks

Mix networks (Chaum, 1981) offer these features:

- › consist of several instances (mixes)
- › each mix
 - › possesses a public key pair,
 - › collects,
 - › decrypts, and
 - › reorders
- › messages and sends n messages
 - › out in one batch
- › Result: No adversary can trace the path of a message



Mix Networks

Sender anonymity

Sender

- › encrypts message N with public key e_R of the receiver
- › chooses n Mixes (M_1, \dots, M_n)
- › encrypts successively with public key e_{M_i} of each mix chosen
- › sends result to first mix



Mix Networks

Construction of a sender's message N'

$$N_{n+1} = e_R(N)$$

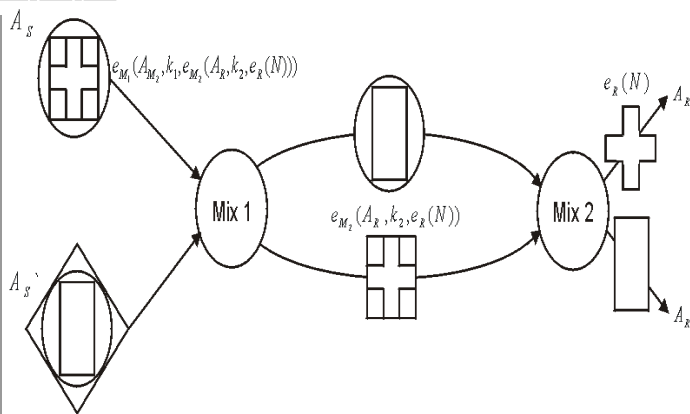
$$N_i = e_{M_i}(A_{M_{i+1}}, k_i, N_{i+1})$$

$$N' = N_n$$

$$A_{M_{n+1}} = A_R$$



Mix Networks



Mix Networks

Receiver anonymity

- › Sender wants to receive an answer
- › **untraceable/anonymous return address**

$$anRA_{n+1} = A_S$$

$$anRA_i = e_{M_i}(anRA_{i+1}, A_{M_i}, k_i)$$

$$anRA(S) = anRA_n$$

- › Sender anonymity and receiver anonymity can be combined



Receiver/Server Anonymity

- Using the anonymising service *Rewebber*
- › <http://www.sec2002.eun.eg/> becomes the **anonymous URL**
 - › **http:**
//www.rewebber.com/surf_encrypted/MT
CIZviiwqq7HNPB2NYEzTC5y7LfWEdZd3
4HCiQiOAGOANNjl3b8J\$eKKnxh5\$
bLidZIWS\$BDAtWcba43Pmrukjsg43vpj\$
1mEwloImDrcHUWf5chzCBuNmrqz2LRT
8os+A=



Mix Networks

- Mix networks are vulnerable for **replay attacks**:
- › Same messages at the mix's input are matched to the same messages at the output
 - › **Correlation attack** is possible
 - › Solution
 - › Each mix must not transport a message or handle an anonymous address twice
 - › Each mix provides a database of already used messages/addresses



Problems of Real Bilateral Anonymous Communication in the WWW

1. Web servers have to publish an enormous amount of anonymous URLs (because of anti-reply mechanism)
2. Anonymous URLs are lengthy and difficult to handle
3. Lack of confidentiality because of the non-existence of authenticated keys



Solution

- › Usage of identity-based cryptosystems (here: **pseudonym-based signature**)
- › Extension of the traditional mix network architecture
 - › Introduction of two instances
 - › **Key Generation Center (KGC)**
 - › **Address Generator (AG)**



Identity Based Cryptosystems (Shamir 1984)/ Pseudonym Based Signature

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

- › Eliminates need of exchanging keys
- › Public key is identity i of a user
- › Secret key g is generated by a trusted party:

$$g^e = i \pmod n$$

$$n = p * q, \text{gcd}(e, \phi(n)) = 1$$
- › e is selected randomly
- › Furthermore f , a hash function, is necessary
- › All parameters, except p and q are known to all instances of the architecture



Identity Based Cryptosystems (Shamir 1984)/ Pseudonym Based Signature

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

- ### Signature
- › User chooses r randomly and computes **signature pair (s, t)**

$$s = g * r^{f(t, m)} \pmod n$$

$$t = r^e \pmod n$$

Verification

$$s^e = i * t^{f(t, m)} \pmod n$$

- › **Signature with pseudonym psd : $g_{psd}(m)$**



Proposed (Extended) Architecture

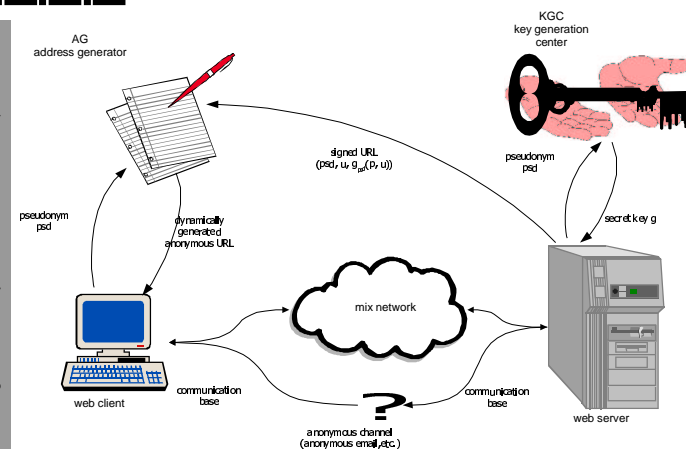
T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

- › Context: WWW, but generally valid for all kinds of communication
- › Components
 - › Group of web clients: want to communicate confidential and to stay anonymous and untraceable
 - › Group of web servers: dito, but additionally want to publish authenticated but still anonymous (pseudonymous) credentials
- › Key Generation Center
- › Address Generator: Produces anonymous URLs
- › (Underlying) Mix network



Proposed (Extended) Architecture

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks





Phases of Operation

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

1. Generation of a secret key g for a given pseudonym psd
2. Sending of a pseudonymous signed URL u to the AG
3. Distribution of a signed communication base
4. Request for/generation of a dynamically built anonymous URL
5. Establishing of communication between client and server
6. Further communication



Generation of a Secret Key g for a Given Pseudonym psd

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

- Provider of web content
- > wants to publish anonymously
 - > selects appropriate pseudonym psd for a cleartext URL u
 - > sends psd to KGC
- KGC
- > calculates corresponding secret key g_{psd}
 - Communication between server/provider and KGC is encrypted
 - KGC does not need to know server/provider necessarily.



Sending of a Pseudonymous Signed URL u to the AG

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

- Server
- > uses URL u as start page
 - > signs u and psd : $(psd, u, g_{psd}(psd, u))$
- AG
- > verifies signature
 - > adds u to its internal list of anonymous URLs



Distribution of a Signed Communication Base

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

- Server builds and distributes a **Communication Base**:
- $$(psd, p, cmt, g_{psd}(psd, p, cmt))$$
- > p : public key used by a client for sending back a chosen session key
 - > cmt : expressive comment
- Distribution by anonymous remailers, etc.



Request for and Generation of a Dynamically Built Anonymous URL

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

Client

- › knows communication base of a server
- › and sends request (containing psd) to AG

AG

- › knows URL corresponding to psd
- › chooses n Mixes and n blinding values randomly
- › constructs anonymous URL:

$$anRA(u) = (A_{M_1}, k_1, e_{M_1}(\dots e_{M_{r-1}}(A_{M_r}, k_r, e_{M_r}(u))\dots))$$



Establishing Communication Between Client and Server and Exchanging a Session Key

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

Client

- › selects (symmetric) session key k_c
- › adds client identification number c_{id} and token CE to his request
- › encrypts all with p from communication base
- › request is sent using $anRA(u)$ and another anonymous return address pointing to the client c , $anRA(c)$

$$[p(k_c, c_{id}, CE)]$$

$$anRA(u) || [p(k_c, c_{id}, CE)] || anRA(c)$$



Communication Between Client and Server Using the Dynamically Built Anonymous URL

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

Server

- › reacts with requested web page using $anRA(c)$
- › encrypts the web page with k_c
- › additionally sends a new anonymous URL (also for links in the web page)
- › For not being identifiable by the server in further communication, the client applies sender anonymity using the anonymous URLs.
- › For authenticated encryption session key k_c ist used.



Problems Solved!

T. Demuth
Establishing Bilateral Anonymous Communication in Open Networks

1. New anonymous URL for each new contact (for avoiding blocking of used addresses)
 - › Using AG for construction of „fresh“ anonymous URLs
2. Problem of unhandy URL
 - › Using pseudonyms easy to handle
3. Authenticated keys for confidential communication
 - › Using a pseudonymously signed communication base



Further Anonymity Considerations

- › If server wants to stay anonymous against KGC
 - › using mix network for communication using sender anonymity and providing an anonymous return address for KGC's response
- › Client wants to stay anonymous against AG
 - › dito



Future Work

- › Single point of failure: AG
 - › knows URL in clear
- › Solution
 - › Introduction of a group of peer AGs
 - › Anonymous distribution of URL as **shared secrets** to AG group
 - › Synchronised generation of dynamically built address parts by the AGs using **privacy homomorphisms**
 - › Composition of the parts by the client



Thank you

for your attention!