

Home Page

Title Page

Contents

« « » »

◀ ▶

Page 1 of 26

Go Back

Full Screen

Close

Quit

Constant Round Authenticated Group Key Agreement from General Assumptions

Emmanuel Bresson

Cryptology Department, CELAR, France

Dario Catalano

École normale supérieure, France

Home Page

Title Page

Contents

« « » »

◀ ▶

Page 2 of 26

Go Back

Full Screen

Close

Quit

Contents

Introduction	3
Our results	7
Security analysis	18
Efficiency	25
Conclusion	26

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 3 of 26

Go Back

Full Screen

Close

Quit

Introduction

Group key agreement schemes allow several parties to come up with a common secret.

These schemes are thus useful in many distributed applications where multicast security (secrecy or integrity) is needed.

To become fully usable, such schemes must offer:

- efficiency (scalability)
- provable security

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 4 of 26

Go Back

Full Screen

Close

Quit

Motivation of our work

Design practical, secure schemes according to some well-defined notions

- Efficiency
 - bits exchanged
 - bit operations performed
 - complexity communication rounds
- Security
 - no party can fix or bias the value of the key (contributory)
 - no outside adversary can get information on the key (privacy)

Motivation (2)

From a more (but not only) theoretical point of view:
Most of the schemes known so far derive from the famous Diffie-Hellman key exchange

Challenge: How to be based on alternative complexity assumptions?

reducing dependence on a complexity assumptions is a line of basic research

- both theoretical (complexity, security point of view)...
- ... and practical importance (implementation issues)

Related work

- Previously proposed schemes
 - Diffie-Hellman extended to groups [AST98, STW96, BCP02]
 - Constant-round scheme for 3 parties [J00]
 - Constant-round DH for groups [BD94, KY03]
- Security models for AKE (Authenticated Key Exchange)
 - Initiated by Bellare *et al.* [BR93, BCK98]
 - Extended for (dynamic) group security [BCP01]

Home Page
Title Page
Contents

« »

◀ ▶

Page 7 of 26

Go Back

Full Screen

Close

Quit

Our results

The scheme proposed

- runs in a constant number of rounds
 - efficient for large groups or slow network
 - still not fully scalable (linear computation)
- offers an alternative to DDH constructions
 - instanciable with RSA-Paillier scheme
- is provably secure under standard assumptions
 - existence of trapdoor permutations

Home Page
Title Page
Contents

« »

◀ ▶

Page 8 of 26

Go Back

Full Screen

Close

Quit

Players and network

- Players have certified public keys
 - a trusted PKI is assumed,
 - messages can be authenticated via signatures;
- network is public and asynchronous
 - provides multi-send capabilities (not necessarily broadcast),
 - is under adversary's control (modification, delay, insertion of messages),
 - players are available through queries made by the adversary.

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 9 of 26

Go Back

Full Screen

Close

Quit

How to deal with active adversaries

- Modular construction of cryptographic protocols
 - Consider protocol in presence of a passive adversary
 - Prove security against such an adversary
 - Use a compiler to transform the (passively secure) protocol into an actively secure one [KY03]
- Allows to consider only passive adversary
 - Design of the protocol is simplified
 - Security proof is much easier and less prone to errors

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 10 of 26

Go Back

Full Screen

Close

Quit

Security definitions

A group key agreement protocol has to satisfy the following:

- Completeness
 - If there is no adversary, the protocol establishes a common key for all P_i
- Semantic security of the session key
 - The session key should be indistinguishable from a random string
- Authentication
 - A key confirmation mechanism prevents \mathcal{A} to fool some players
- Perfect forward secrecy
 - Security of previous session keys remain even if corruption

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 11 of 26

Go Back

Full Screen

Close

Quit

Queries

The following queries are made available to \mathcal{A}

- **Send-query**: allows \mathcal{A} to send arbitrary messages to players (active adversary having entire control over the network)
- **Execute-query**: provides \mathcal{A} with transcript of honest executions (passive eavesdropper)
- **Reveal-query**: used to attack a particular execution (a.k.a. *known key attacks*)
- **Corrupt-query**: models corruption of long-term secret (used to deal with forward-secrecy)
- **Test-query**: the game of semantic security: \mathcal{A} is given either the actual key or a random one

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 12 of 26

Go Back

Full Screen

Close

Quit

Main ideas of the scheme

- Each player sends a contribution nonce
 - A rushing player may wait for other contributions before choosing its own
- Use polynomial secret sharing to distribute information-theoretically hidden masks
 - Separate the protocols in two rounds to ensure every body has sent its nonce
 - Masks are sent (and interpolated) in the second round only
- Protects against *honest-but-curious* players

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 13 of 26

Go Back

Full Screen

Close

Quit

Tools needed 1/3

■ Digital signatures

- A Key Generation algorithm: $\mathcal{KG}(1^\ell) \rightarrow (sk, pk)$
- A Signing algorithm: $\mathcal{S}(sk, m) \rightarrow \sigma$
- A Verification algorithm: $\mathcal{V}(pk, m, \sigma) \rightarrow \{0, 1\}$

■ Secure in the sense of [GMR88]

- Existentially unforgeable
- Under Adaptive Chosen Message Attack
- In the standard computational setting

■ Examples are [CS99], [GHR99]

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 14 of 26

Go Back

Full Screen

Close

Quit

Tools needed 2/3

■ Semantically secure encryption

- A Key Generation algorithm: $\mathcal{KG}(1^\ell) \rightarrow (sk, pk)$
- An Encryption algorithm: $\mathcal{E}(pk, m) \rightarrow c$
- A Decryption algorithm: $\mathcal{D}(sk, c) \rightarrow m$

■ Secure in the IND-CPA sense

- Indistinguishability of ciphertexts [GM84]
- Under Chosen-Plaintext Attack

■ Example: El Gamal (DDH assumption). Chosen in [BC04] for efficiency reason

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 15 of 26

Go Back

Full Screen

Close

Quit

Tools needed 3/3

■ Pseudo-random function families [GGM86]

- A samplable set of functions $\mathcal{F} = \{F_k\}_{k \in \mathcal{K}}$
- These functions are efficiently computable
- Given oracle access to one of them, one can not tell whether sampled from the family or at complete random

■ Used in our construction

- For key confirmation round
- Avoids Random Oracle (OWFs are sufficient)

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 16 of 26

Go Back

Full Screen

Close

Quit

Description of the scheme

■ Public parameters

- A prime p of length ℓ and a pseudo-random function family \mathcal{F} .

■ Round 1

- Each player P_i chooses a random nonce $s_i \in \mathbb{Z}_p^*$ as its own contribution
- P_i chooses at random a $(n - 1)$ -degree polynomial $f_i(z)$ in \mathbb{Z}_p , such that $f_i(0) = s_i$
- Each P_i sends to player P_j the value $f_i(j)$

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 17 of 26

Go Back

Full Screen

Close

Quit

Description of the scheme

- Round 2
 - Each player P_i computes $f(i) = \sum_j f_j(i) \pmod{p}$
 - P_i encrypts the result using the public key of every other P_j
 - Each P_i sends the ciphertexts to intended players
- Confirmation round
 - Upon receiving all ciphertexts, player P_i interpolates privately the polynomial f and sets $\sigma_i = f(0)$ as a seed
 - Each P_i broadcasts $sk_i = F_{\sigma_i}(i)$
 - If all received broadcasts are consistent, set $sk = F_{\sigma_i}(\mathcal{ID})$

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 18 of 26

Go Back

Full Screen

Close

Quit

Security analysis

Theorem 1 *If trapdoor permutations exist, the protocol presented is a secure Group Key Agreement (according to above definitions)*

- Reduction to the semantic security of the scheme under CPA attack
- Reduction to the pseudo-randomness of the family \mathcal{F}
- Reduction to the signature scheme is used through the compiler

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 19 of 26

Go Back

Full Screen

Close

Quit

Main ideas of the proof

- Indistinguishability of the key
- Observe that round 1 gives no information at all
 - The shares $f_i(i)$ are never disclosed
 - All polynomials f_i remain hidden (even in presence of rushing)
- Show that the global shares $f(i)$ are computationally hidden
 - All shares are encrypted in round 2
 - Use hybrid argument to show indistinguishability in such multi-user setting [BBM00]

Home Page

Title Page

Contents

◀ ▶

◀ ▶

Page 20 of 26

Go Back

Full Screen

Close

Quit

- Experiments G_l for $l = 1$ through n
 - Generate polynomials f_1, \dots, f_n and use them in round 1
 - Generate a polynomial \hat{f}_1 s.t. $f_1(1) \neq \hat{f}_1(1)$ and $f_1(j) = \hat{f}_1(j)$ for $j > 1$
 - In round 2, player P_1 sends to P_k an encryption of $f(1)$ if $k > l$ and an encryption of $\hat{f}(1) = \hat{f}_1(1) + \sum_{j>1} f_j(1)$ if $k \leq l$
 - The adversary is challenged on either $F_{f(0)}(P_i)$ or $F_r(P_i)$
- G_0 is the real attack
 - The polynomial \hat{f}_1 is never used, all encryptions in round 2 are performed on $f(1) = \sum_j f_j(1)$
 - We denote by ϵ the advantage of the adversary \mathcal{A}
- G_n is a trivial experiment
 - The polynomial \hat{f}_1 is used in all encryptions in round 2, while the value $f(1) = \sum_j f_j(1)$ is never used, except for answering (half of time) the challenge
 - The polynomial f_1 can be chosen at the very end (and outside of \mathcal{A} 's view)

Relation among hybrid games

- Reduction to semantic security of the encryption scheme
 - Choose to messages m_0 and m_1
 - Get as a challenge an encryption c of m_b where $b \in_R \{0, 1\}$
- Show what happens between intermediate games
 - Choose at random an index $j \in [2, n]$
 - Define for P_1 polynomials f_1 and \hat{f}_1 in such a way that $f(1) = m_0$ and $\hat{f}(1) = m_1$ (remind $f_1(i) = \hat{f}_1(i)$ for $i > 1$)
 - In round 2, encrypt m_1 for players P_2, \dots, P_{j-1} and encrypt m_0 for players P_{j+1}, \dots, P_n . Give c to P_j
- We get the differential between two consecutive experiments
 - If $b = 0$, we are playing G_{j-1} while if $b = 1$, we are playing G_j
 - We can deduce:
$$\left| \Pr[\mathcal{A}(\dots) = 1 | G_j] - \Pr[\mathcal{A}(\dots) = 1 | G_{j-1}] \right| \leq \text{Adv}_{ENC}^{\text{ind}}(\mathcal{A})$$

Concluding the proof

- By definition of the advantage:
$$\left| \Pr[\mathcal{A}(\dots) = 1 | G_1] - \Pr[\mathcal{A}(\dots) = 1 | G_n] \right| = \frac{1}{2} \text{Adv}^{\text{ake}}(\mathcal{A})$$
- Conditioning on the choice of j , one gets
$$\text{Adv}^{\text{ake}}(\mathcal{A}) \leq 2(n-1) \times \text{Adv}_{ENC}^{\text{ind}}(\mathcal{A})$$
- Advantage on the pseudo-random function family has to be considered as well
 - Consider an experiment where F is replaced by a purely random function
 - Adversary \mathcal{A} has clearly no advantage
 - Experiments differs by a quantity $\leq \text{Adv}^{\text{prf}}(\mathcal{F})$

A closer look on assumptions

- Trapdoor permutation family $\{f_n\}_n$
 - A generator $\mathcal{G}(1^\ell) \rightarrow (n, t_n)$, where t_n is the trapdoor
 - An efficient computation algorithm $(n, x) \rightarrow f_n(x)$
 - An efficient inversion algorithm $(n, f_n(x), t_n) \rightarrow x$
 - No efficient inversion algorithm $(n, f_n(x)) \rightarrow x$ without the secret
- Trapdoor permutations are sufficient for encryption schemes
- One-way functions are necessary and sufficient for signatures
- One-way functions imply existence of PRF's

Application: RSA-based construction

- The RSA-Paillier cryptosystem [CGHN01]
 - Let $N = pq$ and consider the group $\mathbb{Z}_{N^2}^*$
 - Choose e s.t. $(e, \lambda(N^2)) = 1$
 - Encrypt $m \in \mathbb{Z}_N$ as $c = r^e(1 + mN) \bmod N^2$, for $r \in_R \mathbb{Z}_N^*$
 - Decryption is done by recovering $r = \sqrt[e]{c} \bmod N$
 - The above transformation $(r, m) \rightarrow c$ is a trapdoor permutation over $\mathbb{Z}_{N^2}^* \approx \mathbb{Z}_N^* \times \mathbb{Z}_N$
 - The scheme is semantically secure under the *Decisional Small e -Residues Assumption* (and one-way under RSA)
- Our construction can be applied to this particular cryptosystem

Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Page 25 of 26

Go Back

Full Screen

Close

Quit

Efficiency

- Encryption is slower than traditional RSA mod N
 - The gap can be reduced using FFT
- Decryption is essentially dominated by the first step $r = \sqrt[e]{c} \bmod N$ and is thus comparable to RSA mod N
 - Moreover the security still holds for small e
- The resulting construction for key agreement. . .
 - $n - 1$ exponentiations mod N^2 to encrypt and $n - 1$ exponentiations mod N to decrypt
 - Using $e = 3$, this reduces to roughly $2(n - 1)$ exp. mod N
 - The cost of signatures is to be added for active security

Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Page 26 of 26

Go Back

Full Screen

Close

Quit

Conclusion

- We have constructed practical scheme
 - based on El Gamal (and, for this sole reason, DDH)
 - based on RSA-Paillier cryptosystem (thus, non a DDH)
 - both schemes requires a constant number of rounds and are fairly efficient
- From the theoretical point of view
 - we generalize our previous construction to be based on general assumption (existence of trapdoor permutations)
 - we proceed in the standard model
 - we gain security against active security using [KY03]