

Design of Secure Key Establishment Protocols: Successes, Failures and Prospects

Colin Boyd

Information Security Research Centre

Queensland University of Technology

Invited Talk for Indocrypt 2004

Outline

1. Key Establishment
2. Bellare–Rogaway model
3. Canetti–Krawczyk model
4. Comparing formal methods and cryptographic paradigms

Key Establishment

Accept a key suitable for secure communications between two or more parties. Mandatory properties:

- key must be new (freshness)
- key must be confidential (key authentication)
- protocol must be efficient and effective

Some Protocols Types

- Server-based key transport (Needham–Schroeder 1978)
- Key agreement based on Diffie–Hellman (STS 1992)
- Password-based protocols (Bellovin–Merritt 1992, IEEE 2004)
- Group key agreement (Ingemarsson 1982, Burmester–Desmedt 1994)

Some Optional Properties

- Entity authentication
- Key confirmation
- Forward secrecy
- Resistance to key compromise impersonation
- Anonymity of principals
- Deniability

Example: STS Protocol

Digital signatures add authentication to the well-known Diffie–Hellman protocol

Exponents r_A and r_B are chosen randomly by A and B respectively and are used to form the session key $K_{AB} = g^{r_A r_B}$

1. $A \rightarrow B : g^{r_A}$
2. $B \rightarrow A : g^{r_B}, \{Sig_B(g^{r_B}, g^{r_A})\}_{K_{AB}}$
3. $A \rightarrow B : \{Sig_A(g^{r_A}, g^{r_B})\}_{K_{AB}}$

Focus of this talk

Key establishment is a fundamental problem for security

- How sure are we that protocols are secure?
- What are some of the main proof models used?
- How can we improve confidence in, and understanding of, security proofs?

Proofs in the Cryptography Community

- Reductionist approach to proof: if adversary can break protocol then some computational assumption is false
- Three distinct models:
 1. Bellare–Rogaway (1993, 1995, 2000)
 2. Canetti–Krawczyk (1998, 2001)
 3. Shoup (1999)
- How many proven secure protocols are there? See the Provably Secure Protocols Lounge:
<http://sky.fit.qut.edu.au/~choo/lounge.html>

Outline

1. Key Establishment
2. Bellare–Rogaway model
3. Canetti–Krawczyk model
4. Comparing formal methods and cryptographic paradigms

Bellare–Rogaway Model

- Developed by Bellare and Rogaway (1993, 1995)
- Many protocols proven secure in this model
- Adversary controls communications:
 - chooses which principals send messages
 - can change messages to anything it can compute
 - can obtain any session key or long-term key
- Each principal U runs an unbounded number of instances Π_U^s indexed by s

Queries available to adversary

Four queries are available to the adversary in the original model

$\text{Send}(U, s, M)$	Send message M to instance Π_U^s
$\text{Reveal}(U, s)$	Reveal session key (if any) accepted by Π_U^s
$\text{Corrupt}(U, K)$	Reveal state of U and set long-term key of U to K
$\text{Test}(U, s)$	Obtain either the session key accepted by Π_U^s or a random string

Only one Test query can be issued.

Definition of Security

- A *target session* is determined by the Test query
- Protocol is secure if adversary cannot distinguish between the target session key and a random string
- The instance targetted in the Test query must not have been the subject of Reveal or Corrupt
- The *partner* of the instance targetted in the Test query must not have been the subject of Reveal or Corrupt

Extensions to Basic Model

Basic model ensures:

- key authentication
- key freshness

Bellare, Pointcheval and Rogaway (2000):

- added Execute query to deal with password guessing
- adapted Corrupt query to deal with forward secrecy
- suggested a definition of partnering based on session ID

Other extensions are certainly possible

Bellare–Rogaway 3PKD Protocol (1995)

- Server-based key transport
- Needham-Schroeder style
- Server S shares long-term keys with all users
- S chooses new session key and sends to any two users on request

Bellare–Rogaway 3PKD Protocol (1995)

1. $A \rightarrow B : N_A$
2. $B \rightarrow S : N_A, N_B$
3. $S \rightarrow A : E_{K_{AS}}(K_{AB}), \text{MAC}_{K'_{AS}}(A, B, N_A, E_{K_{AS}}(K_{AB}))$
4. $S \rightarrow B : E_{K_{BS}}(K_{AB}), \text{MAC}_{K'_{BS}}(A, B, N_B, E_{K_{BS}}(K_{AB}))$

Keys K_{AS} and K'_{AS} are shared between A and S and assumed independent

Similar for K_{BS} and K'_{BS}

Proof for 3PKD

3PKD was proven secure by Bellare and Rogaway assuming:

1. secure MAC is used;
2. encryption provides indistinguishability under chosen plaintext attack.

In other words, if an efficient adversary can distinguish the target session key from a random key with probability better than guessing then one of the above assumptions is *false*

Partnering Difficulties in 3PKD

- A crucial part of security proof is notion of *partnering*
- Proof of 3PKD uses complex partnering function instead of *session ID*
- There is no reasonable way to define session ID for 3PKD
- Simple change to protocol allows straightforward session ID and simplifies proof of security [CBHM04b]

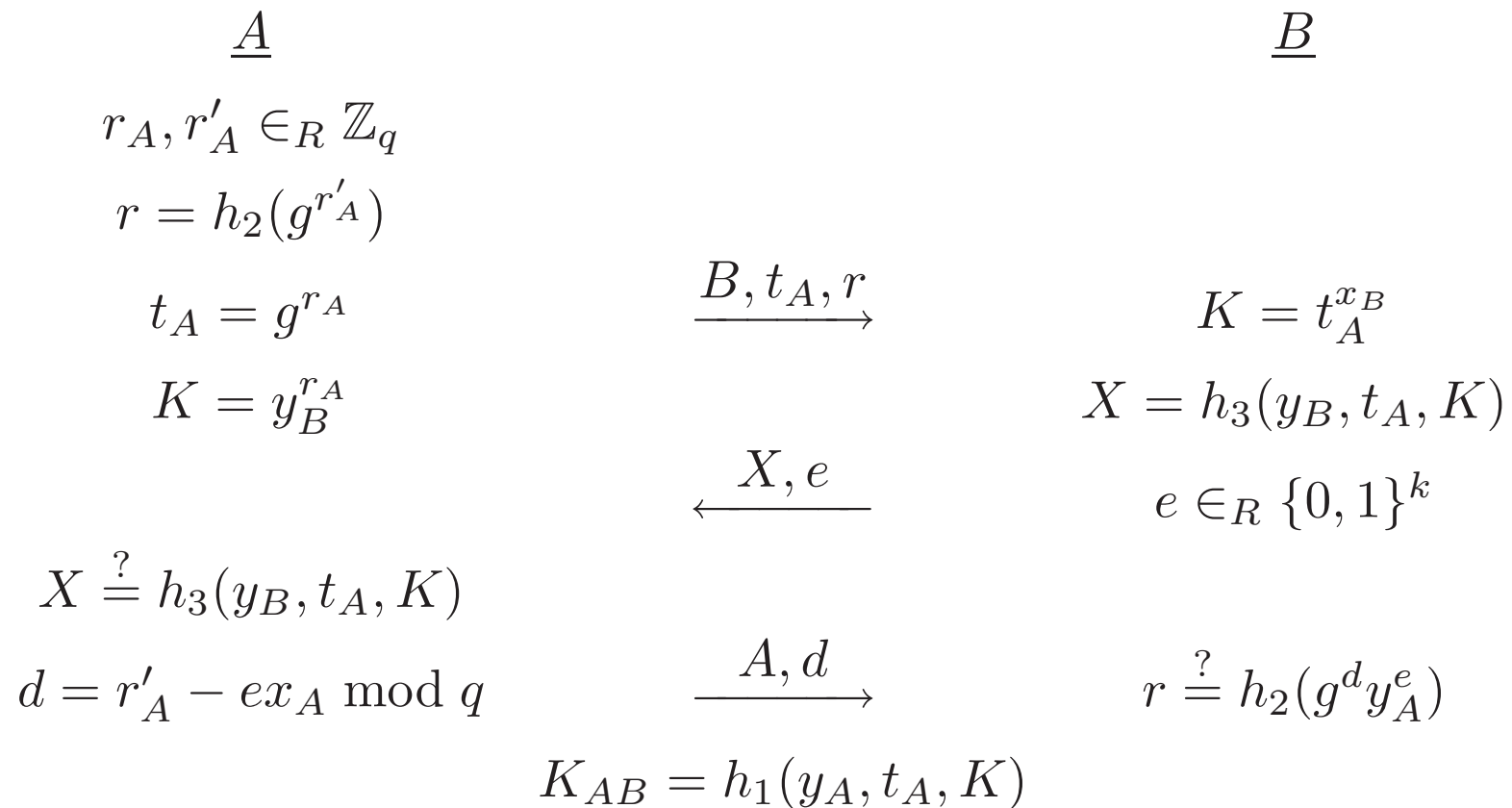
Modified Bellare–Rogaway 3PKD Protocol

1. $A \rightarrow B : N_A$
 2. $B \rightarrow S : N_A, N_B$
 3. $S \rightarrow A : E_{K_{AS}}(K_{AB}), MAC_{K'_{AS}}(A, B, N_A, N_B, E_{K_{AS}}(K_{AB}))$
 4. $S \rightarrow B : E_{K_{BS}}(K_{AB}), MAC_{K'_{BS}}(A, B, N_A, N_B, E_{K_{BS}}(K_{AB}))$
- Session ID can be defined as $N_A \parallel N_B$
 - Also proven secure in Bellare–Rogaway (2000) model

Jakobsson–Pointcheval Protocol

- Key agreement for lower power devices
- Proven secure in Bellare–Rogaway model (Financial Cryptography 2001)
- Wong and Chan found simple error (Asiacrypt 2001)
- Protocol fixed by Jakobsson and Pointcheval but Wong and Chan proposed ‘improved’ protocol with claimed proof of security
- Improved protocol also broken

Original Jakobsson–Pointcheval Protocol



Signature (r, e, d) is independent of shared key

Limitations of Bellare–Rogaway approach

- Proofs are usually long and complex:
 - often skipped by readers
 - may have errors
- Proofs are monolithic and fragile:
 - not helpful for protocol design
 - difficult to re-use proofs

Bellare–Rogaway Model Summary

- Successfully used by many researchers
- Simplified partnering was a step forward
- Proofs still very complex

Outline

1. Key Establishment
2. Bellare–Rogaway model
3. Canetti–Krawczyk model
4. Comparing formal methods and cryptographic paradigms

Canetti and Krawczyk's Modular Approach

- Bellare, Canetti and Krawczyk (1998) introduced the idea of a *modular* approach to protocol proofs
- Original 1998 version used simulatability definition of security — too strong to be practical
- Canetti and Krawczyk (2001) recast the model with a security definition based on indistinguishability

Two worlds

Ideal world: Adversary cannot fabricate or divert messages

Real world: Adversary can send any message that it can compute

- *Authenticators* are compilers used to transform protocols secure in the ideal world into protocols secure in the real world
- A general result of Canetti and Krawczyk guarantees that the resulting real-world protocol is secure
- As in Bellare–Rogaway model, adversary chooses which entities to activate and has the ability to reveal session keys and corrupt parties

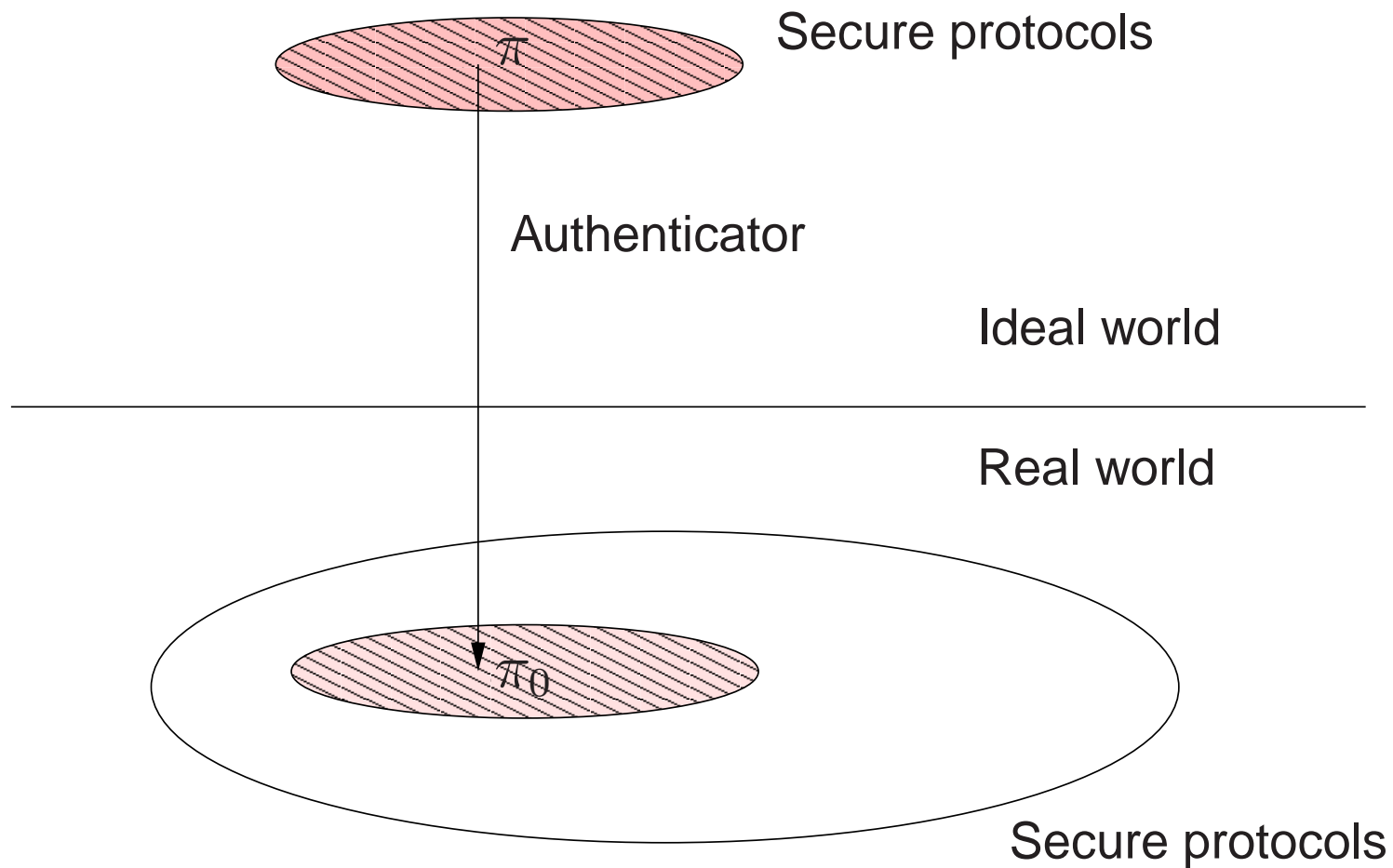


Figure 1: Using an authenticator to compile protocols

Current Known Ideal World Protocols

Protocol	Computational assumption
Diffie-Hellman	DDH
Key transport	CCA-secure encryption
El-Gamal type	Gap-DH
Server-based key transport	CPA-secure encryption
Tripartite Diffie-Hellman	BDH

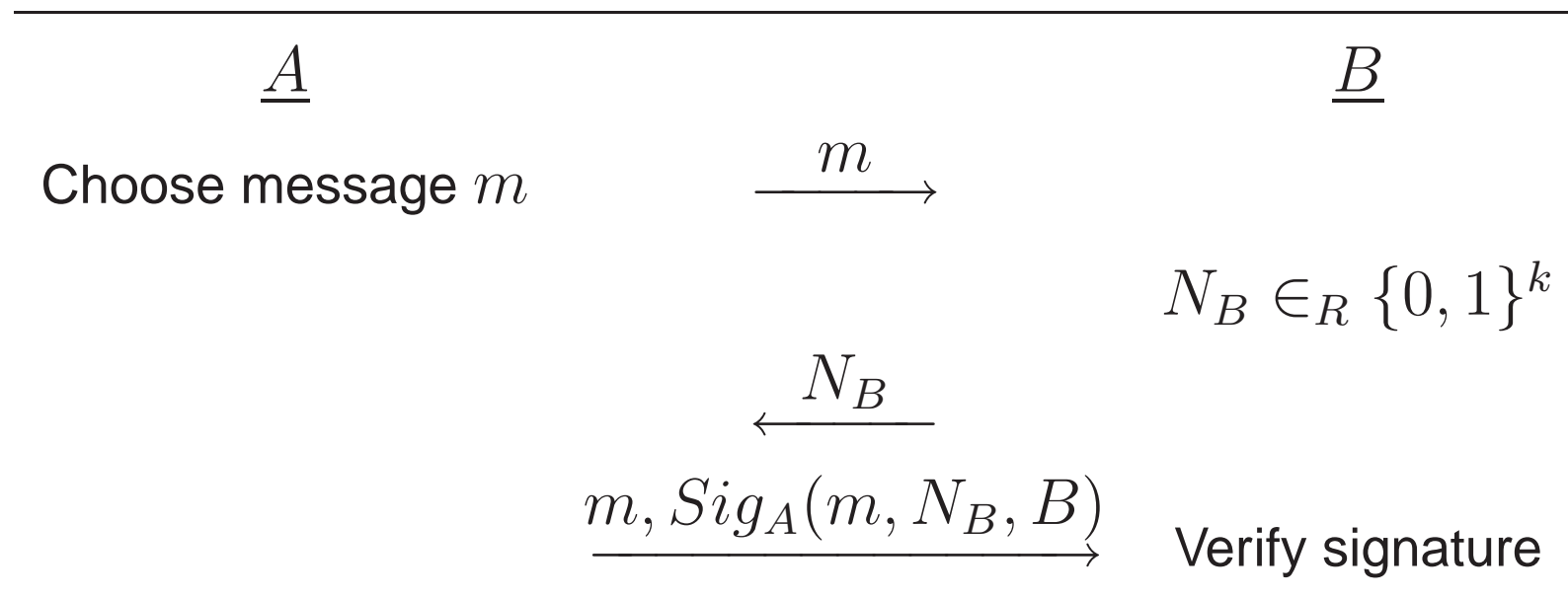
Current Known Authenticators

Authenticators	Computational assumption
Signature-based	Secure signature
Encryption-based	Secure MAC and CCA-secure encryption
MAC with existing shared key	Secure MAC
Password-based	CCA-secure encryption
Statically keyed	CDH and random hash function

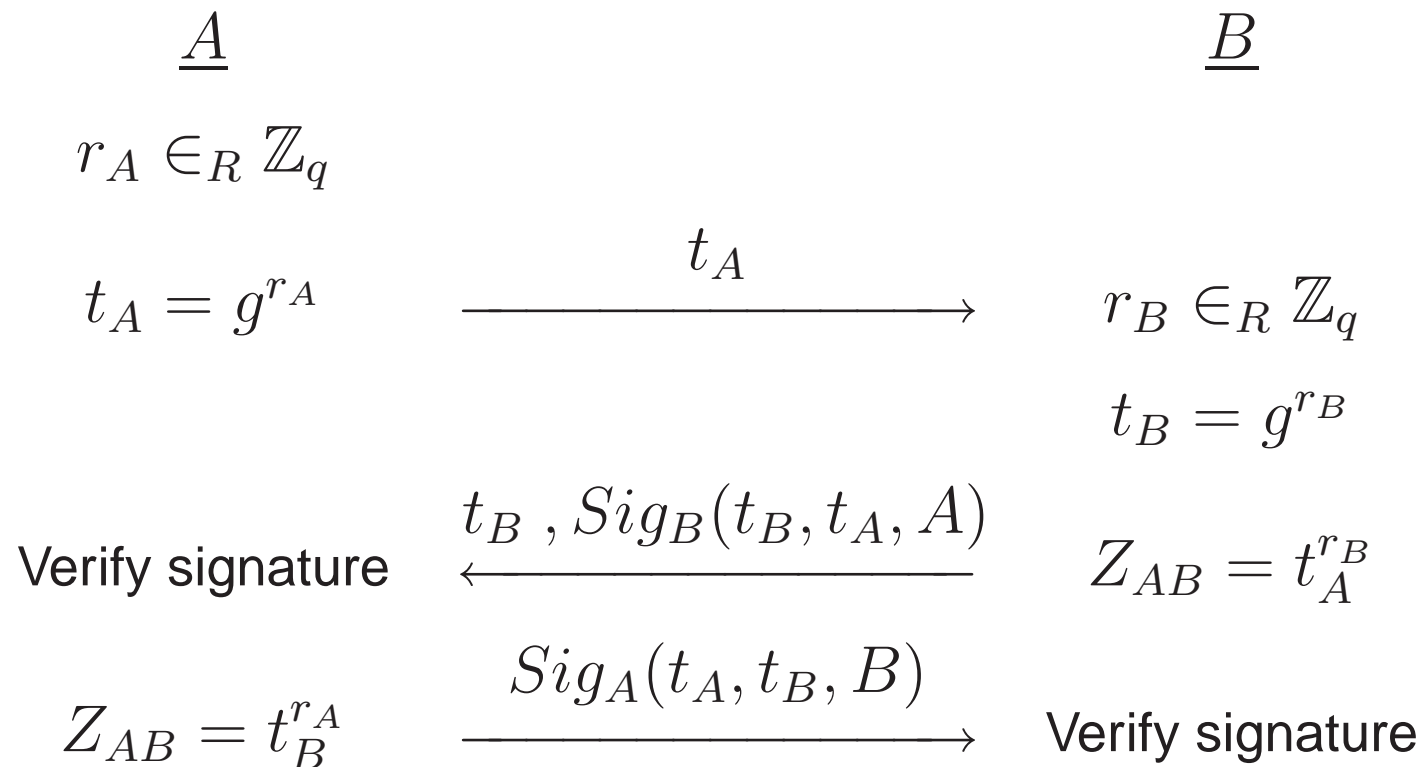
An Example

- Protocol in ideal world is plain Diffie–Hellman
- Authenticator is based on any secure signature
- Apply authenticator to each message and then optimise

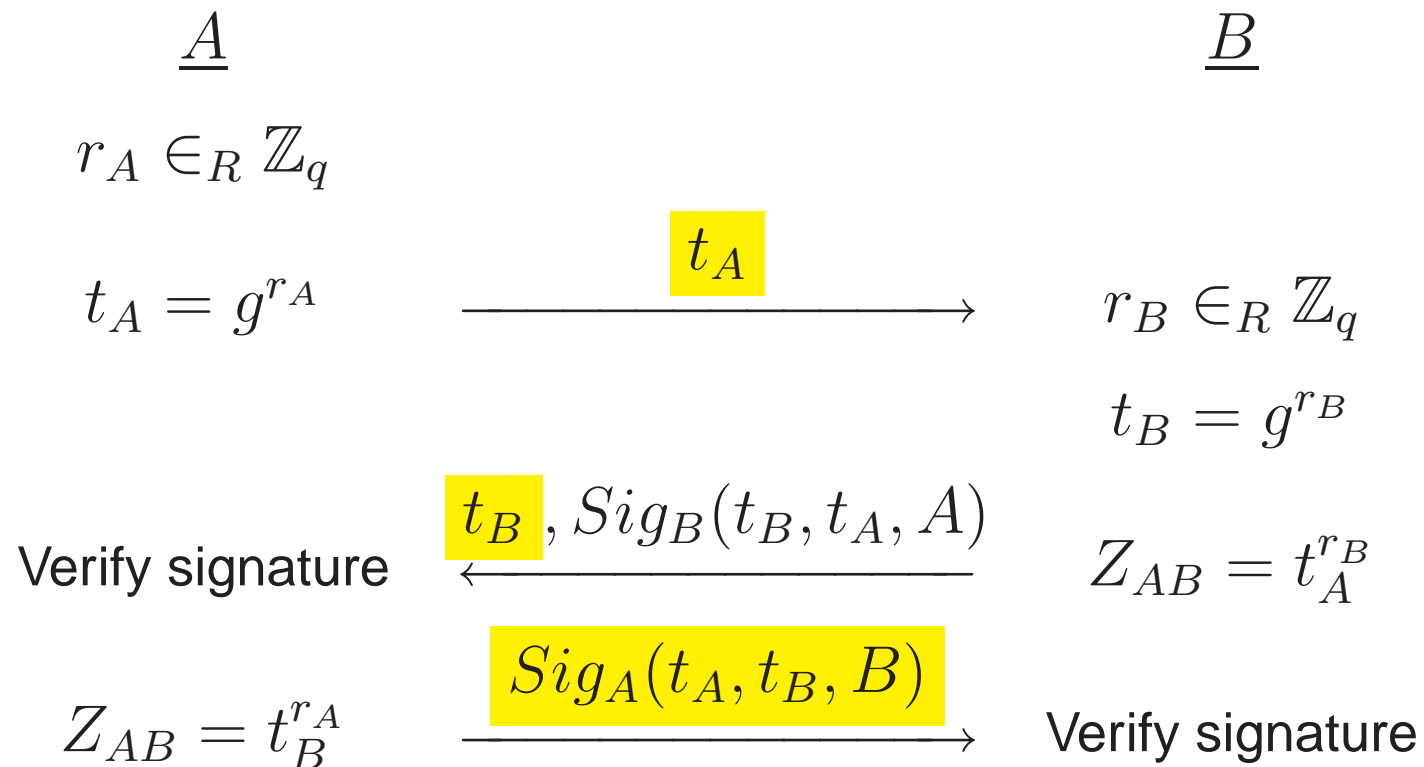
Signature-based authenticator



Secure protocol in the real world



Secure protocol in the real world



New (Unpublished) Results for Optimisation

- Mixed authenticators can be used
- Messages can be interleaved in most obvious cases
- Repeated plaintext elements can be deleted
- Nonce values may be overloaded
- Session identifiers may be constructed after protocol starts

Limitations of the Canetti–Krawczyk Approach

1. Some protocols do not have any reasonable decomposition into an ideal world protocol and authenticator
2. The range of protocols that can be captured is limited to key exchange protocols plus protocols for secure channels using exchanged keys

Canetti–Krawczyk Model Summary

- Modular approach is a step towards design
- Proofs are shorter, but still hard for humans
- Not all protocols can be reached

Outline

1. Key Establishment
2. Bellare–Rogaway model
3. Canetti–Krawczyk model
4. Comparing formal methods and cryptographic paradigms

Two Protocol Communities

1. Formal methods community

-
-
-

2. Cryptography community

-
-
-

Two Protocol Communities

1. Formal methods community

- Tradition from computer security
- Search for insecure states in model
- Meadows, Millen, Lowe, Paulson, . . .

2. Cryptography community

-
-
-

Two Protocol Communities

1. Formal methods community

- Tradition from computer security
- Search for insecure states in model
- Meadows, Millen, Lowe, Paulson, . . .

2. Cryptography community

- Tradition from computational complexity
- Reduce to better known computational problem
- Bellare–Rogaway, Shoup, Canneti–Krawczyk, . . .

Typical differences between approaches

Formal methods	Cryptography
Automated tools	Hand proofs

Typical differences between approaches

Formal methods	Cryptography
Automated tools	Hand proofs
Black-box cryptography	Detailed model of cryptography

Typical differences between approaches

Formal methods	Cryptography
Automated tools	Hand proofs
Black-box cryptography	Detailed model of cryptography
Deterministic model	Probabilistic model

Typical differences between approaches

Formal methods	Cryptography
Automated tools	Hand proofs
Black-box cryptography	Detailed model of cryptography
Deterministic model	Probabilistic model
Security by absence of attack	Defined security properties

Typical differences between approaches

Formal methods	Cryptography
Automated tools	Hand proofs
Black-box cryptography	Detailed model of cryptography
Deterministic model	Probabilistic model
Security by absence of attack	Defined security properties
Penetrator	Adversary

Lowe's Attack on STS Protocol

- Lowe published an attack in 1996 on STS
- Adversary plays in middle between A and B
- Attack only 'visible' if identities are added to messages
- A 'thinks' protocol is completed with B but B does not know A exists
- An attack only in certain models

Lowe's Attack on STS

$$1. \quad A \rightarrow C_B : \quad A, B, g^{r_A}$$

$$1'. \quad C \rightarrow B : \quad C, B, g^{r_A}$$

$$2'. \quad B \rightarrow C : \quad B, C, g^{r_B}, \{Sig_B(g^{r_B}, g^{r_A})\}_{K_{AB}}$$

$$2. \quad C_B \rightarrow A : \quad B, A, g^y, \{Sig_B(g^{r_B}, g^{r_A})\}_{K_{AB}}$$

$$3. \quad A \rightarrow C_B : \quad A, B, \{Sig_A(g^{r_A}, g^{r_B})\}_{K_{AB}}$$

Strengths of formal methods

- automatic tools allows fast and error-free analysis
- recent work of Datta, Derek, Mitchell and Pavlovic (CSFW 2004) allows protocol design via templates

Weaknesses of formal methods

- cryptography is not modelled accurately
- no formal definition of what security means

Strengths and weaknesses are *complementary* to those in cryptographic approach

Towards Unifying the Two Worlds

Simple strategies

- Use same model for computational proof and for machine analysis [CBHM04a]
 - Would complementary checking have found original problem in Jakobsson-Pointcheval protocol?
- Formal specification and static checking of simulations
 - Experience shows that simulations are frequently specified with omissions or illegal actions

Ambitious strategies

- Faithful abstraction of cryptographic primitives to replace black-box cryptography
 - Canetti's universal composability
 - Backes–Waidner–Pfitzmann library

Future Challenges

- New protocol requirements will arise
-
-
-
-
-

Future Challenges

- New protocol requirements will arise
- Dealing with denial of service
-
-
-
-

Future Challenges

- New protocol requirements will arise
- Dealing with denial of service
- **Modelling side-channel attacks**
-
-
-

Future Challenges

- New protocol requirements will arise
- Dealing with denial of service
- Modelling side-channel attacks
- Automatic analysis with faithful cryptography
-
-

Future Challenges

- New protocol requirements will arise
- Dealing with denial of service
- Modelling side-channel attacks
- Automatic analysis with faithful cryptography
- **Secure design with chosen properties**
-

Future Challenges

- New protocol requirements will arise
- Dealing with denial of service
- Modelling side-channel attacks
- Automatic analysis with faithful cryptography
- Secure design with chosen properties
- **Automatic checking of computational proofs**

Summary

- Cryptographic proofs have greatly increased understanding and reliability of key establishment protocols
- Proofs are difficult for humans to handle
- The way forward should include machine support
- Still many challenges ahead