Quantum Key Distribution in the Classical Authenticated Key Exchange Framework

Michele Mosca, Douglas Stebila,
Berkant Ustaoglu

University of Waterloo; Perimeter Institute
Queensland University of Technology
Izmir Institute of Technology

arXiv:1206.6150

QKD in classical authenticated key exchange framework

- State-of-the-art in classical key agreement models
- What secrets can be leaked while keeping the session key secure?
  - monolithic information leakage >>> fine-grained leakage
- Modeling QKD in this framework
  - using computational or information-theoretic authentication

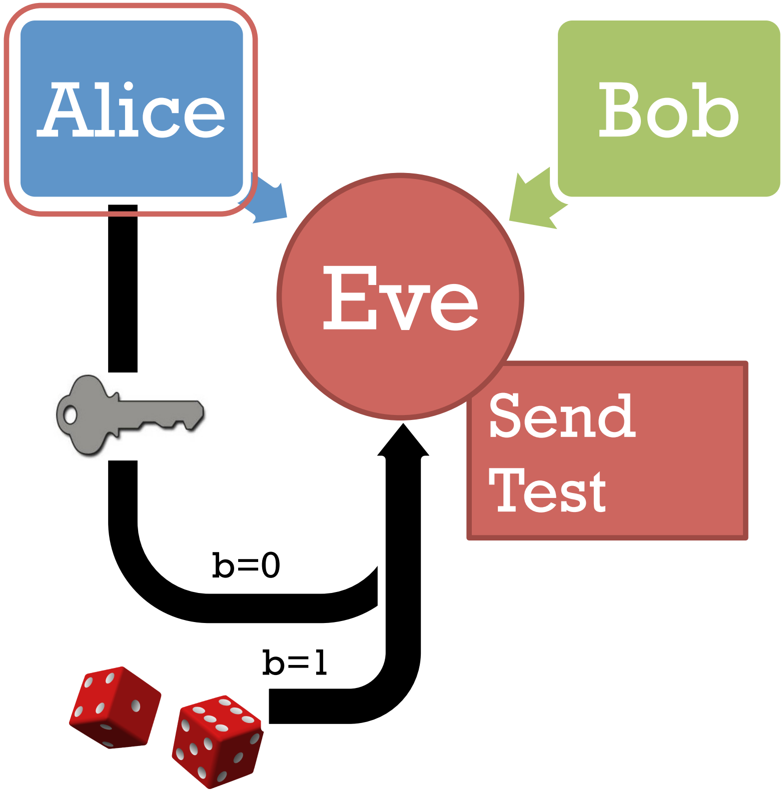
Authenticated key exchange

- Two parties establish a shared secret using only public communication and an authenticated channel
- Classical public-key key exchange protocols:
  - Diffie–Hellman (1976)
  - Key transport using public key encryption (e.g. RSA) (1978)
- QKD: BB84, EPR, Time-reversed, …

Provable security

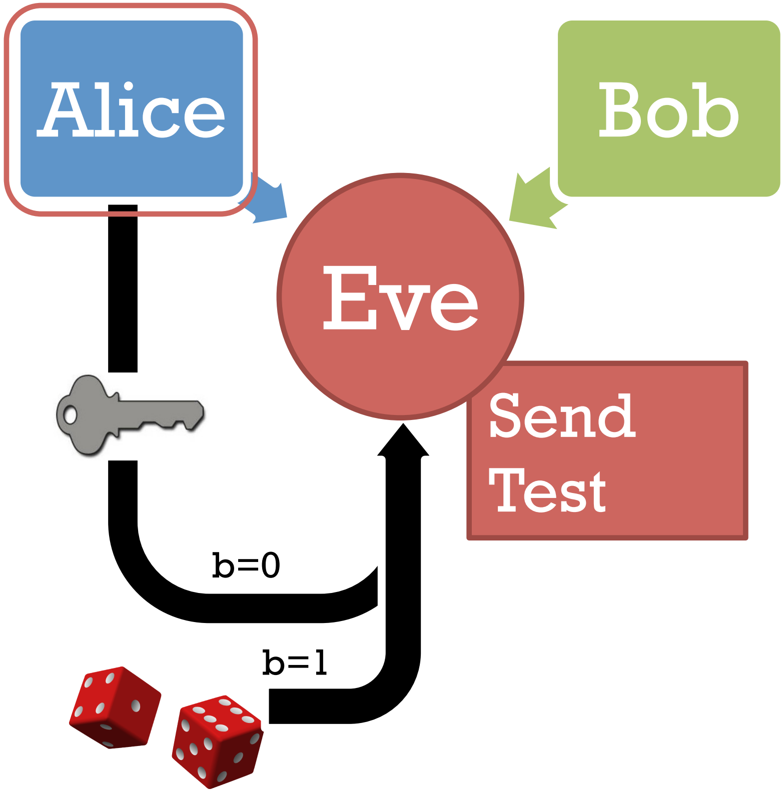
- Provable security introduced by Goldwasser and Micali for public key encryption in 1984.
- A primitive or protocol is a tuple of algorithms.
- A security property (or "security model") is described by an interactive algorithm between a challenger and an adversary algorithm.
- Security result is a bound on the probability a particular class of algorithms can cause the challenger to output 1.

Simple security model



- Two parties, Alice and Bob execute a **session** of a protocol
- **Send:** Eve controls all communication between parties.
- **Test:** Eve picks a target session.
  Challenger flips a coin b.
  If b=0: give Eve real key
  If b=1: give Eve random string
- **Eve's goal: guess b (decide if the Test session's key was real or random).**

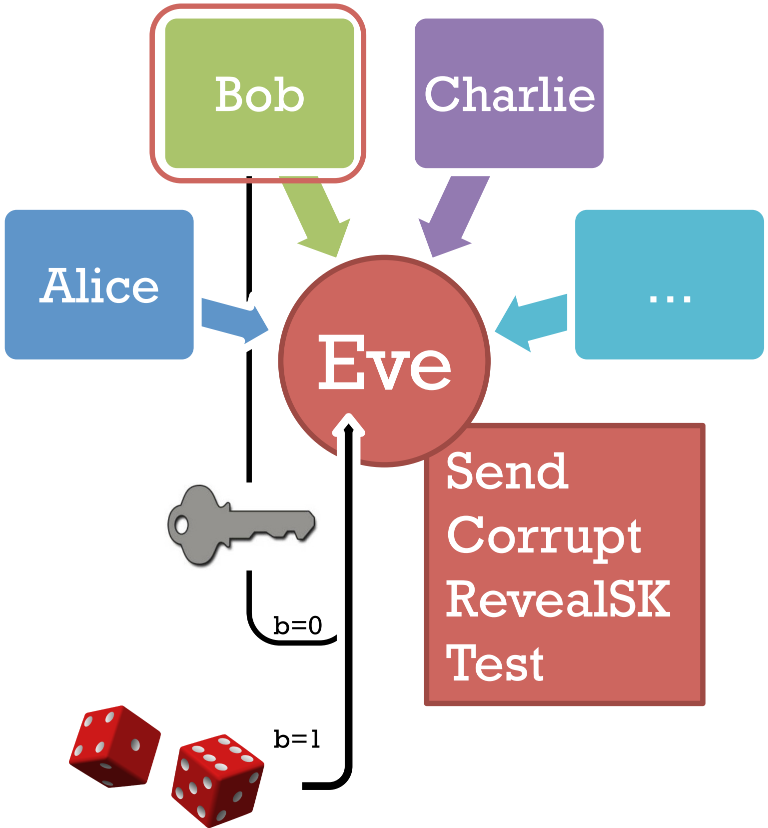
Simple security model



Limitations

- Only 2 parties
- Only 1 session
- No information leakage allowed

BR93/BJM97 security model



- Multiple parties execute many sessions
- Two parties, Alice and Bob execute a **session** of a protocol
- **Send:** Eve controls all communication between parties.
- **Corrupt:** Eve can learn long-term secret keys
- **RevealSessionKey**
- **Test:** Eve picks a target session.
  Challenger flips a coin b.
  If b=0: give Eve real key
  If b=1: give Eve random string
- Eve's goal: guess b (provided that the session was fresh a.k.a. uncorrupted)

Fresh sessions in BR93/BJM97

- If Eve can reveal session keys and corrupt long term keys, which sessions ought to remain secure?

- A session π at party A is **fresh** if
  - No Corrupt(A)
  - No SessionKeyReveal(π)
  - No Corrupt(B) where B is the peer of A
  - No SessionKeyReveal(π’) where π’ is a **matching session** to π

Matching session: (incomplete) transcripts match

Signed Diffie–Hellman protocol

Alice

- Long-term key
  $(pk_a, sk_a) \leftarrow \text{Sig.KeyGen}()$
  Obtain pk_b

1. $x \xleftarrow{\$} \{1, \ldots, p-1\}$
   $X \leftarrow g^x$
   $\sigma_A \leftarrow \text{Sig.Sign}(sk_a, X)$
2. $\text{Sig.Verify}(pk_B, Y, \sigma_B)$
   $k_{AB} \leftarrow H(Y^x)$

$\xrightarrow{X,\ \sigma_A}$

$\xleftarrow{Y,\ \sigma_B}$

Bob

- Long-term key
  $(pk_b, sk_b) \leftarrow \text{Sig.KeyGen}()$
  Obtain pk_a

1. $y \xleftarrow{\$} \{1, \ldots, p-1\}$
   $Y \leftarrow g^y$
   $\sigma_B \leftarrow \text{Sig.Sign}(sk_b, Y)$
2. $\text{Sig.Verify}(pk_A, X, \sigma_A)$
   $k_{AB} \leftarrow H(X^y)$

Not secure if **ephemeral key** ever revealed.

What if the randomness used in a session is leaked?

- Not reasonable to assume that Alice's computer is perfect, even if there's a wall around it.
- Weak randomness generation
  - Early versions of Netscape's PRNG were poorly seeded [Goldberg, Wagner 1995]
  - Debian's version of OpenSSL discarded most of the entropy used in PRNG [Bello 2008]
- PC compromised by spyware/ malware
- Can we still achieve security even with weak randomness?

MQV-style protocols

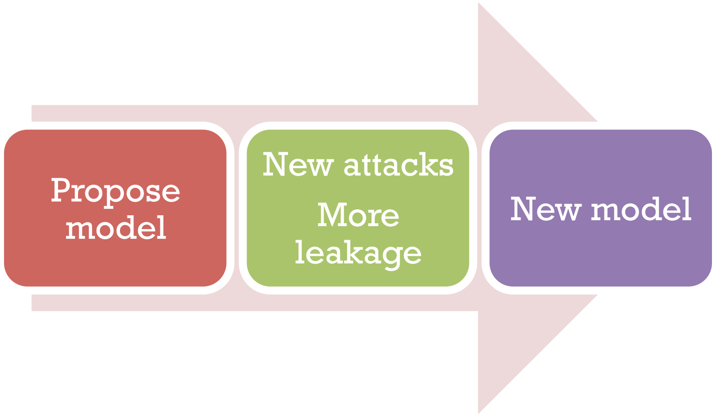
MQV, HMQV, NAXOS, CMQV, UP, SF, …

Alice		Bob
- Long-term key $a \leftarrow\$ \{1, \ldots, p-1\}$ $A \leftarrow g^a$ Obtain pk_b		- Long-term key $b \leftarrow\$ \{1, \ldots, p-1\}$ $B \leftarrow g^b$ Obtain pk_a
1. $x \leftarrow\$ \{1, \ldots, p-1\}$ $X \leftarrow g^x$	$\xrightarrow{X}$	1. $y \leftarrow\$ \{1, \ldots, p-1\}$ $Y \leftarrow g^y$
2. $Z1 \leftarrow (YB^{H(X)})^{x+a}$ $Z2 \leftarrow (YB)^{x+H(Y)a}$ $k \leftarrow H(Z1, Z2, Alice, Bob, X, Y)$	$\xleftarrow{Y}$	2. $Z1 \leftarrow (XA)^{y+H(Y)b}$ $Z2 \leftarrow (XA^{H(X)})^{y+b}$ $k \leftarrow H(Z1, Z2, Alice, Bob, X, Y)$

Secure even if at most one, but **not both,** of a party's session key and ephemeral key revealed after protocol completion

Security models for key exchange

- **BR93:** Bellare-Rogaway (1993)
- Blake-Wilson–Johnson–Menezes (1997)
- Bellare–Pointcheval–Rogaway (2000)
- **CK01:** Canetti–Krawczyk (2001)
- CK_HMQV: Krawczyk (2005)
- **eCK:** LaMacchia–Lauter–Mityagin (2007)



Composability?

- Vast majority of key exchange papers use "direct" security models with no composability theorems.
- CK02: UC version of CK01
- CHKLM05: weak corruptions only

Comparison of security models

Newer models add more adversarial powers to model more information leakage.

	BR93/BJM97	CK01	eCK
Send control all communication	✔	✔	✔
Corrupt learn long-term secret key	✔	✔	✔
SessionStateReveal reveal internal state of party	✖	✔	✖
EphemeralKeyReveal learn short-term randomness	✖	✖	✔
SessionKeyReveal learn session keys	✔	✔	✔

Which is the best model?

- **BR93/BJM97**
  - Doesn't allow leakage of any ephemeral secrets
- **CK01**
  - SessionStateReveal is sometimes ambiguously defined
  - Attacks: key compromise impersonation
- **eCK**
  - EphemeralKeyReveal can't be called before session begins
  - Can play "tricks" to achieve somewhat unnatural security

- CK01 and eCK formally and practically incomparable. [Cremers 2010]
- None include the "wider" scope of a real-world protocol such as certification/key registration, (re-)negotiation, …
- Still a matter of debate as to the most appropriate definition(s) to use.
  - eCK-like models most widely used

Existing QKD security models

Stand-alone definitions

- Only two parties (+ Eve)
- Assume authentication

Universal composability definition

Ben-Or, Horodecki, Leung, Mayers, Oppenheim (TCC 2005)

- In simplified version of Ben-Or-Mayers composability framework
- No information leakage
- Information-theoretic authentication

Definitions compatible with simulatability & composability frameworks

- e.g. Renner 2005

Quantum composability frameworks

- Ben-Or, Mayers 2004
- Fehr, Schaffner 2008
- Unruh 2004, 2009/10
- Maurer, Renner 20??

QKD in the language of classical authenticated key exchange

Goal

- Develop a unified security model that can be used to describe the security of:
  - Classical authenticated key agreement protocols
  - QKD with information-theoretic authentication
  - QKD with computationally secure authentication

Benefits

- Directly compare qualitative properties of various classical and quantum AKE protocols
- QKD as a standard cryptographic primitive
- Formalization of "folklore" result that QKD with computational authentication is long-term secure as long as not broken before protocol completes
  [various position papers]
  [Müller-Quade, Unruh 2010]

Prepare-send-measure QKD

BB84

six-state protocol

- Randomness:
  - Long-term authentication key
  - Basis choices
  - Data bits
  - Information reconciliation randomness
  - Privacy amplification randomness

Measure-only QKD

Ekert91

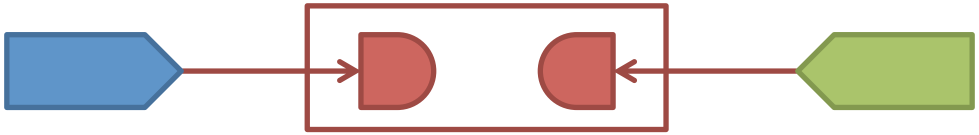
BBM92

- Randomness:
  - Long-term authentication key
  - Basis choices
  - Information reconciliation randomness
  - Privacy amplification randomness

Prepare-send-only QKD

Time-reversed [BHM96, Ina02]

Measurement device-independent [LCQ12, BP12]



- Randomness:
  - Long-term authentication key
  - Basis choices
  - Data bits
  - Information reconciliation randomness
  - Privacy amplification randomness

Unified security model

- Multiple parties execute many sessions
- Two parties, Alice and Bob execute a **session** of a protocol
- **SendC, SendQ:** Eve controls all communication between parties.
- **Partner:** Eve can learn long-term keys or randomness
- **RevealNext:** Eve can learn randomness before it's used
- **Test:** Eve picks a target session.
  Challenger flips a coin b.
  If b=0: give Eve real key
  If b=1: give Eve random string
- Eve's goal: guess b (provided that the session was fresh)
- Session output specifies freshness condition

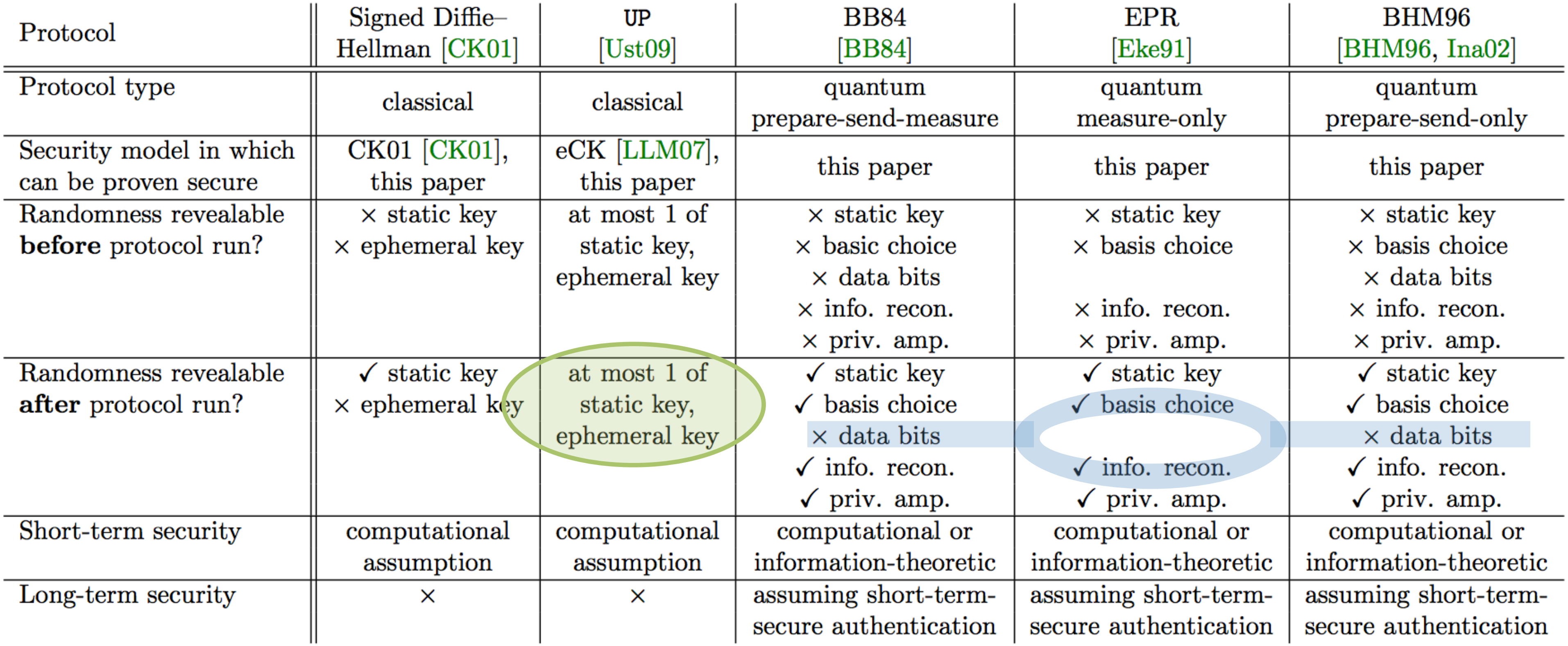
Adversary types

- **Short-term security**: Bounds on Eve:
  - t_c: classical runtime
  - t_q: quantum runtime
  - m_q: quantum memory
- **Long-term security**:
  1. (t_c, t_q, m_q)-bounded Eve_1 interacts with the protocol to produce a cq transcript
  2. Unbounded quantum Eve_2 operates on transcript

- Can interpolate from
  - purely classical Eve: $t_c = \text{poly}, t_q = 0, m_q = 0$
  - reasonable upper bound on today's quantum Eve: $t_c = \text{poly}, t_q = 10^3, m_q = 10^3$
  - poly quantum Eve: $t_q = \text{poly}(\lambda), m_q = \text{poly}(\lambda)$
  - unbounded quantum Eve: $t_q = \infty, m_q = \infty$

Mirrors UC framework long-term security definitions of Müller-Quade and Unruh (2010).

Protocol comparison



Protocol	Signed Diffie–Hellman [CK01]	UP [Ust09]	BB84 [BB84]	EPR [Eke91]	BHM96 [BHM96, Ina02]
Protocol type	classical	classical	quantum prepare-send-measure	quantum measure-only	quantum prepare-send-only
Security model in which can be proven secure	CK01 [CK01], this paper	eCK [LLM07], this paper	this paper	this paper	this paper
Randomness revealable **before** protocol run?	× static key × ephemeral key	at most 1 of static key, ephemeral key	× static key × basic choice × data bits × info. recon. × priv. amp.	× static key × basis choice × info. recon. × priv. amp.	× static key × basis choice × data bits × info. recon. × priv. amp.
Randomness revealable **after** protocol run?	✓ static key × ephemeral key	at most 1 of static key, ephemeral key	✓ static key ✓ basis choice × data bits ✓ info. recon. ✓ priv. amp.	✓ static key ✓ basis choice ✓ info. recon. ✓ priv. amp.	✓ static key ✓ basis choice × data bits ✓ info. recon. ✓ priv. amp.
Short-term security	computational assumption	computational assumption	computational or information-theoretic	computational or information-theoretic	computational or information-theoretic
Long-term security	×	×	assuming short-term-secure authentication	assuming short-term-secure authentication	assuming short-term-secure authentication

Questions for QKD

- Design MQV-style prepare-and-send protocol secure even when data bits are revealed
    - Maybe only computationally secure in that case
- Leakage-resilient cryptography provides more fine-grained description of information leakage
    - e.g. reveal arbitrary function f(x) of internal state x, where |f(x)| bounded per session or overall
    - Prove security of QKD against a class of leakage functions, then argue that side-channels in a real-world protocol are modeled by that class of leakage functions